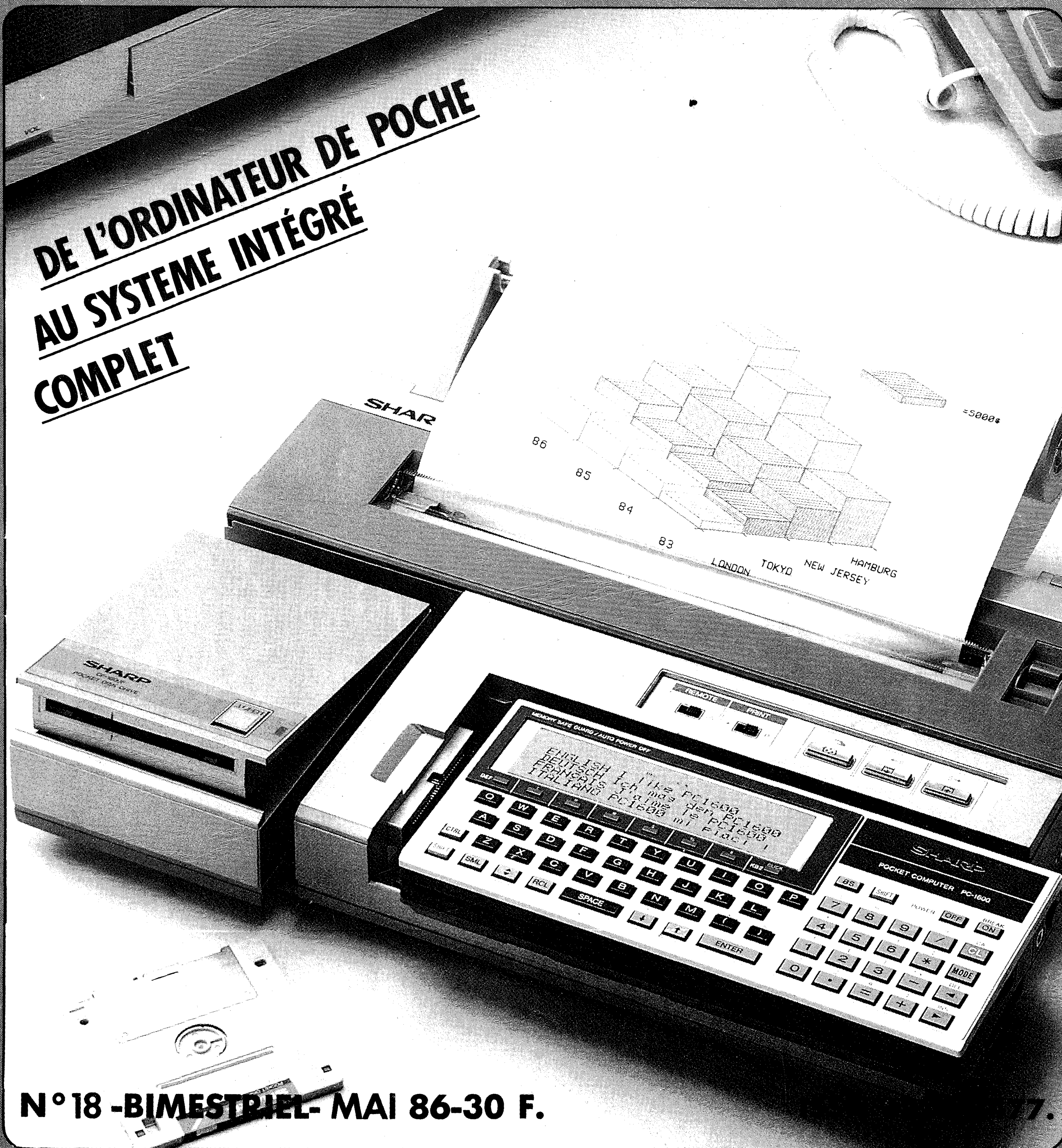
La revue des

SHARPENTIERS



DE L'ORDINATEUR DE POCHE AU SYSTEME INTÉGRÉ COMPLET

N° 18 -BIMESTRIEL- MAI 86-30 F.

LA REVUE DES UTILISATEURS DE MICRO-ORDINATEURS ET POCKET-COMPUTERS SHARP

VOIR NOUVEAUTE
p.6–7

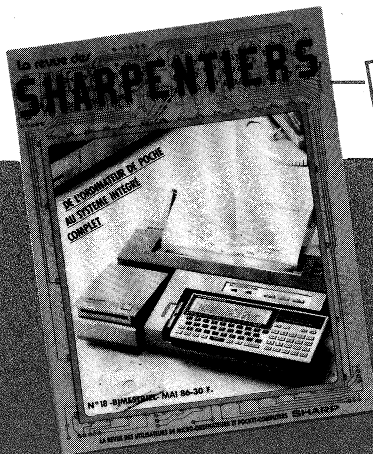


CE-1600P

CE-1600F

CE-1650F

PC-1600



LE SHARPENTIER SOMMAIRE N° 18

INFOS CLUB 4/5
NOUVEAUTÉS 6/7

SYSTEME – APPLICATIONS (8 à 10)

MULTIPOSTE CONTRE RESEAU LOCAL : LE COMBAT FRATRICIDE .... 8 à 10

SHARP ENSEIGNEMENT (11 à 20)

- Système d'équation par la méthode Jacobi 11
- Bensolver 13-14
- Logique mathématique 15
- Pivot de Gauss 16
- Quadrature Gaussienne à 16 points 17
- Fonctions logiques 18-20

BASIC (21 à 25)

BASIC ET RECURSIVITE .... 21 à 24
LES SOUS PROGRAMMES 24/25

PC (26 à 54)

PC 1500
ASTUCES 26/27
SUPER CALL 27
CLASSEMENT 28
INITIATION LM 5801 .... 29 à 31
GALAXY FLY 31/32
A PROPOS DE L'ASSEMBLEUR 33

TOUS LES PC
ASTUCES 33
SYSTEMES ET SOUS PGMS LM 34
INITIATION AU LM ESR-H (4e) 34/35

PC 1211
NAVIGATION ASTRO 36
COMPLEMENT A FONCTION DU N° 16 36

PC 1251
GESTION DE PORTEFEUILLE BOURSIER 37/38

PC 1261
LES PHASES DE LA LUNE .... 39 à 43

PC 1401
CALCUL DE RESSORTS .... 43 à 47

PC 1350
MODE ANGULAIRE 48
DOG FIGHT .... 48 à 53
AMORTISSEMENT 54

MZ (55 à 67)

MZ 700
TABLES TRAÇANTES 55
CALC 700 .... 56 à 60

MZ 700/800
COPY FICHIERS 60

MZ 800
BASIC 800 .... 61/62
MODIFICATIONS AUS BASICS 62/63
DIRECTORY 64
LECTURE VARIABLE 64
DRUNKY .... 64 à 66

MZ 80K
IMPRESSION DE LISTING SOURCE ZEN 66/67

Rédacteur en Chef
Sylvain BIZOIRRE

Rédacteurs
Luc BURELLER
Marc GIRONDOT
Jean-François VIGNAUD

Ont collaboré a ce numéro
Thierry ALLARY
Jacques ANDRE
Frédéric BLONDIAU
Pierre BURCKLE
Xavier BOMBA
Henri CARUSO
Benoit CHAFFANGEON
Jean-Marc DREVET
Henri FAUCON
Maurice GODEAU
Jean HERY
Patrice JACOB
Michel MENARD
Jean MILLET
Alain MORETTE
Martin RAYROLLE
Hervé SZYCHTER
Luis Marchesi ULLASTRES

PUBLICITE :
S. BIZOIRRE
Tél. : 48.34.93.44.

REALISATION :
IN QUARTO
19, rue Frédéric Lemaître
75020 PARIS

BUGS...BUGS...BUGS.

Les « BUGS », ce sont les erreurs de toutes natures découvertes (Trop tard !) dans notre bulletin.

Pour commencer très fort : les bugs des « bugs » du numéro 17;

§ 4 ligne 1
L=TAN x = SIN x / COS x
§ 6 Remplacer tous les blancs par " ".
(Malgré plusieurs tentatives d'initiation de notre imprimeur à l'informatique, celui-ci continue à penser que nous plaçons nos petits signes " $ &" pour faire joli...
Page 19, astuce 2
Input $ ( ), $ ( )
Astuce 3
dans l'expression, Z1 est à remplacer par Z>1
Page 26
1ere. ligne, 2e. colonne : GOSUB NNNN:N=N-P
Page 14 "A la recherche de l'entier"
Colonne 2 : INT A par NOT -A+1
Page 22
WORD LISTER, contrairement à ce qui a été écrit, est tout à fait relogeable.

MZ
Numéro 16 page 62

TEXY
Ajouter ",1100" en fin de lignes 210
Ajouter "GOTO 870" en fin de ligne 850

Numéro 17 page 61

Si vous possédez un MZ 700, il vous faudra modifier le programmme
SAISIE HEXA ainsi :

```
230 IFT 16THENT$="0"+HEX$(T):GOTO240
235 T$=RIGHT$(HEX$(T),2)
```

L'instruction BEEP est à remplacer par MUSIC"A"

Numéro 12 page 18

(Voir corrections CADRAN SOLAIRE)

```
 55 Q=ATN(1/TAN B/K) + W + 180 *(B<0)
 60 FOR G=P TO Q STEP M
 62 A=ATN (K*TAN F)+B
145 S=ATN(*RCOS(G/4-W))
200 PRINT"GRAND 8"
221 N=31*M-31-3*(M>2)-(M>4)-(M>6)-(M⊇>9)-(M>11)
240 FOR F=1 TO 3 : U=(T-O)/Q+E*180/π*SIN U : NEXT F
241 V=2 * ATN (TAN(U/2)*√((1+E)/(1-E)))
```

BULLETIN n° 16

PAGE 29

Considérations temporelles

```
20:CLEAR :DIM A$(0)*78:A$="7F7F7F":A$(0)=A$+A$+A$+A$:A$(0)=A$(0)+A$(0)+A$(0)+A(0):G-PRINT A$(0);A$(0);A$(0);A(0)
```

PAGE 27
Astuce 1
Fin de texte : SPACE INS

PAGE 22

Colonne 2 et ligne 4
Pour introduire l'instruction, soit deux-points soit rien.

PAGE 28

La logique du Polinominal Counter
Colonne 2
Remplacer les + par des
et c+c h par c=c h

PAGE 15

"LINCO" Modifier ligne 1520
1520:IF M(K,P)<> THEN 1550

BULLETIN n° 10

Dans la case 01, il y a SBC (X) et non SBX

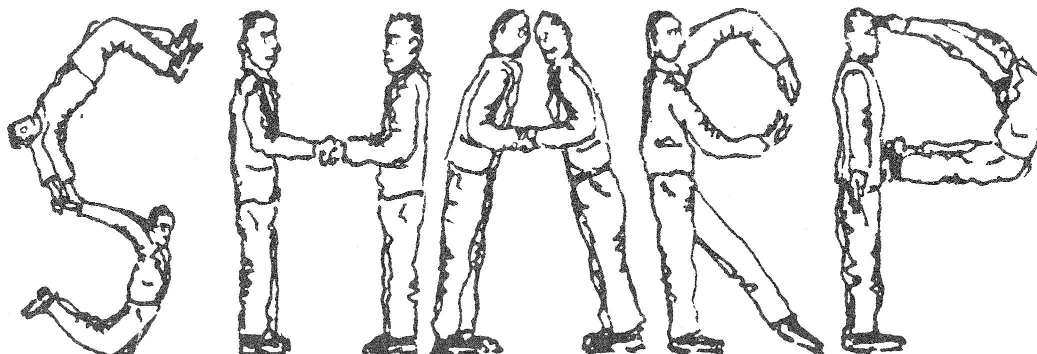
ANNONCES club

Vends PC-1261 10,4 K (85)+interface CE-124 pour magnéto à K7 normales+manuel+K7 de PGMS+revues du club=**1 450 F** à débattre Tél : 69.21.29.66 après 19 H

Vends PC-2500 : 2 800 F+carte 16K : **1 100 F** **Duplanil J.-Pierre - 2, rue Ravel 92300 Levallois Tél : 47.31.45.23**

Vends ordinateur de poche SHARP. PC-1450+interface cassette CE-124 (décembre 85) encore sous garantie. **Prix : 1 200 F.** L. Corbineau - 163, avenue de Brigaode 59650 Villeneuve-d'Ascq. Tél : 20.91.27.59

A vendre SHARP MZ80B 64 K +imprimante à pointe MZ80P5+multimake (cassette-tableaux) **Prix : 4 000 F** (1980). Tél : 46.09.94.11 poste 1154



ET LE BASIC !

C'est l'expression que l'on retrouve dans bon nombre de vos lettres et qui remet régulièrement en question l'éternel problème de l'initiation au BASIC. Nous n'avons, jusqu'à ce jour, jamais entamé de série d'articles d'initiation pour deux raisons essentielles : -1- Le nombre de pages de notre bulletin est limité (finances obligent) et la densité d'informations y est très importante, tellement importante qu'on atteint parfois la limite de lisibilité ! Il nous paraît donc plus opportun d'utiliser cette surface pour la présentation et l'explication d'applications crées par vous et directement exploitables sur vos PC ou MZ. Une initiation au BASIC tenant compte des spécificités de chacune de nos machines occuperait une place importante et nous obligerait à supprimer certaines de ces applications. -2- Il existe sur le marché une bonne trentaine d'ouvrages consacrés exclusivement à l'initiation BASIC. Hormis quelques instructions particulières, les BASIC Sharp sont très proches du BASIC standard dont traitent la majorité de ces ouvrages ; il ne nous parait donc pas utile de réinventer le monde et de réécrire ce qui a été déjà expliqué moulte fois.

Parallèlement à cette initiation, ce numéro du Sharpentier inaugure une nouvelle rubrique intitulée tout simplement "BASIC" qui tentera d'expliquer quelques notions fondamentales de programmation adaptées aux basic's de nos machines et agrémentées d'exemples. Alors, si vous avez trouvé la routine-géniale-inédite-super élaborée-qui-permet-de... n'hésitez pas à nous en faire profiter pour que cette rubrique vive, grâce à vous.

DESSINS, NE PAS CONFONDRE...

Continuez, vos premiers dessins sont superbes et commencent à égayer votre cher bulletin. Petite précision tout de même : nous sommes une revue pour tous publics et certains auteurs dont les œuvres, bien que superbes, avaient un rapport pour le moins lointain avec l'informatique me pardonneront certainement de ne pas les avoir publiés. Bref, pour vos prochaines œuvres, ne confondez pas dessin et des s...s !

PLAGIAT...

Monsieur,

Je vous prie de trouver ci-joint une photocopie d'un programme dont je suis l'auteur, publié dans la revue MICRO 7 d'avril 1983.

Comme il est facile de le vérifier, toute la partie calculs proprement dite, c'est-à-dire la grande majorité des lignes, a été reproduite telle quelle dans le programme THEME ASTRAL de monsieur Philippe PAILLOU publié dans la revue du Club numéro 17.

Le procédé manque de correction ! C'est le moins qu'on puisse dire, car il était facile de me contacter pour accord avant d'adapter mon programme pour une présentation graphique différente... et le faire publier.

Le programme publié dans MICRO 7 me valut un certain courrier ; c'était, pour la plupart, des problèmes d'adaptation à d'autres machines !
Mais des remarques d'astrologues un peu « puristes » m'ont amené à refaire une nouvelle mouture de ce programme dans laquelle les « cupsides » des maisons sont calculés avec toute la précision nécessaire ainsi que les principaux « ASPECTS » qui manquaient dans la version d'avril 1983.
Cette nouvelle version a été publiée dans le numéro Spécial Informatique de la revue MEGAHERTZ de 1984 ; elle a sans doute échappé à M. Paillou...

Pour information et à toutes fin utiles...
Veuillez agréer, Monsieur, l'assurance de ma meilleure considération.

Jean MAZET

Monsieur MAZET ne nous a pas demandé de publier ce courrier, nous avons cependant préféré le faire, par souci d'honnêteté et pour clarifier une situation qui pourrait devenir très dangereuse pour l'avenir de notre bulletin.
Le fait de publier et de faire profiter à des milliers de lecteurs du fruit de ses travaux ou de ses recherches procure toujours un plaisir et une satisfaction personnelle et intellectuelle qui font souvent oublier les nombreuses heures passées dans la mise au point du programme ou de l'article publié. Il est, par contre, très frustrant de retrouver ces mêmes travaux, généralement légèrement modifiés, et signés de la main d'une tierce personne. C'est pourquoi nous comprenons parfaitement le désarroi et le mécontement légitimes de Monsieur MAZET et lui adressons toutes nos excuses.

Au-delà de ce qui ne doit pas être un procès d'intentions, il peut être utile de signaler que nous recevons parfois des articles très documentés qui se révèlent être de pâles copies d'articles déjà pu liés dans d'autres revues.
Comme nous lisons pratiquement toutes les revues d'informatique, nous avons pu éviter cet éceuil... jusqu'au numéro 17.

Pour éviter qu'une telle mésaventure ne se reproduise, ne nous faites parvenir que des programmes ou articles dont vous êtres l'auteur à 100% cela permettra à notre revue de conserver l'originalité, l'indépendance et le caractère unique qui font sa force.

S.B.

PC 1600

LES NOUVEAUTÉS

LES NOUVEAUTÉS

Puissance, Communication et compatibilité, tels sont les 3 maître-mots qui ont présidé à la conception du dernier né et du plus puissant des ordinateurs de poche de la gamme Sharp : Le PC 1600. Ses potentialités lui permettent de traiter l'information à armes égales avec les « gros » ordinateurs de table tout en conservant les deux atouts majeurs d'un « Pocket » : La portabilité et une très grande flexibilité d'emploi.

PRESENTATION

Ce qui frappe, dès la première prise en mains, c'est l'incontestable air de famille avec le grand-frère PC 1500. Dimensions extérieures rigoureusement identiques, même ergonomie du clavier, même « look » dans le choix des couleurs du boitier. Pas de doute, les concepteurs du PC 1600 ont tenu à conserver les avantages pratiques et l'esthétique qui ont fait le succès du PC 1500. A première vue, la seule différence notable, c'est l'afficheur. Il comporte maintenant 4 lignes de 26 caractères et peut afficher 4992 points en mode graphique.

LA MEMOIRE

Là, inutile de rechercher les ressemblances. Dans le PC 1600, c'est la profusion. Jugez plutôt : 96 Ko. de ROM qui contiennent 2 basic's, dont un en mode PC 1500, un système d'exploitation de disquettes (oui ! oui !) et une gestion complète des entrées-sorties et communications externes. Quant à la RAM disponible pour vos programmes et données, elle atteint les 80 Ko. (Quatre-vingt mille octets... pour les sceptiques). Cette mémoire est modulable en fonction de vos besoins. Equipé à l'origine de 16 Ko., le PC 1600 peut recevoir deux modules supplémentaires de 32 Ko. chacun. Comme dans tous les PC, la totalité de la mémoire est non volatile et conserve les informations qu'elle a reçues lorsque la machine est éteinte...

Ainsi gonflé, le PC 1600 peut rivaliser sans complexe avec la grande majorité des micro-ordinateurs dits « de table » tout en sachant rester très discret dans votre poche ou dans un coin de votre malette.



Les 2 logements pour modules mémoire

L'UNITE CENTRALE

Ou plutôt LES unités centrales ; car dans le PC 1600, il y a deux microprocesseurs : Un LH 5803, cousin germain du LH 5801 bien connu des possésseurs de PC 1500 et une version CMOS du très célèbre Z80 qui équipe tous les micros de table de la série MZ. Pourquoi deux microprocesseurs ? D'abord pour conserver la compatibilité logicielle avec la très importante bibliothèque de logiciels du PC 1500 ; ensuite pour obtenir une rapidité de traitement équivalente à celle des gros ordinateurs. Le basic du PC 1600 fonctionne pratiquement trois fois plus rapidement que celui du PC 1500.

UN SYSTEME COMPLET INTEGRE

Le PC 1600 est, en fait, le cœur d'un ensemble comprenant tous les éléments utiles à l'exploitation optimale d'un système informatique complet – Eléments intégrés dans un boitier unique et protégés dans une malette robuste et esthétique.

L'UNITE DE DISQUETTES

Etudiée spécialement pour les ordinateurs de poche, elle utilise des disquettes au format 2,5 pouces et permet le stockage ultra rapide de 128 Ko. (64 Ko. par face) de données ou programmes en mode séquentiel . Le transfert d'informations entre l'unité centrale et la mémoire de masse atteint ainsi un niveau de sécurité et un fiabilité nettement supérieurs au système à cassette ; l'emploi d'un magnétophone restant cependant possible.



Les disquettes au Format 2,5 pouces

L'IMPRIMANTE

Intégrée dans le même boitier, l'imprimante table traçantes CE 1600P permet l'impression de textes ou graphiques sur papier en feuilles ou rouleaux au format A4 (21 x 29,7 cm). Quatre couleurs sont possibles pour l'écriture grâce au système à barillet dont la fiabilité a été largement démontrée sur les différentes imprimantes de la gamme Sharp, de la plus petite (CE 150) à la plus sophistiquée (CE516 P) ; ces deux modèles peuvent d'ailleurs être également connectés au PC 1600.

LA COMMUNICATION

C'est l'un des points fort du PC 1600. Son interface RS 232 intégrée permet le dialogue avec pratiquement tous les systèmes informatiques petits ou grands. La connexion à un modem acoustique permet la télécommunication avec de très nombreuses bases de données. La véritable révolution, dans ce domaine, prend la forme d'un cable anodin, long et fin. Dans ce cable, point de fils de cuivre, mais une fibre optique qui, comme son nom devrait l'indiquer, conduit de la lumière et non de l'électricité. Ou est la révolution ? Tout simplement dans le fait que les transmissions serie peuvent maintenant s'effectuer sur de très grandes distances, à des vitesses ou la traditionnelle transmission électrique s'essouffle (38 400 bauds, dans notre cas), et à l'abri de toutes perturbations électriques dues à un environnement industriel sévère : Hautes tensions, parasites électriques, etc.

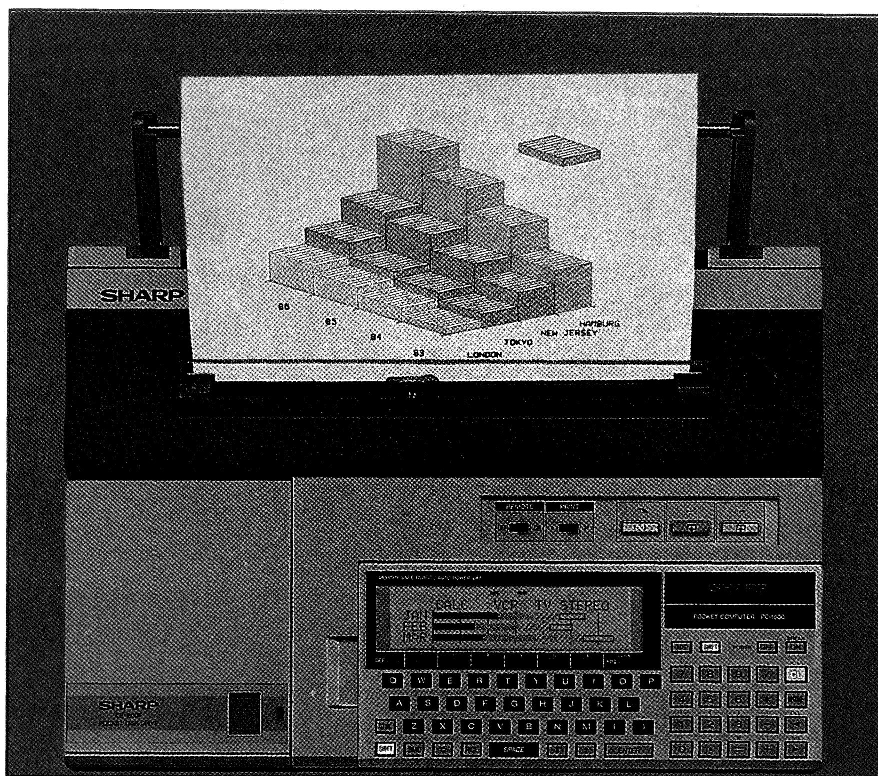
LES APPLICATIONS EN TEMPS REEL

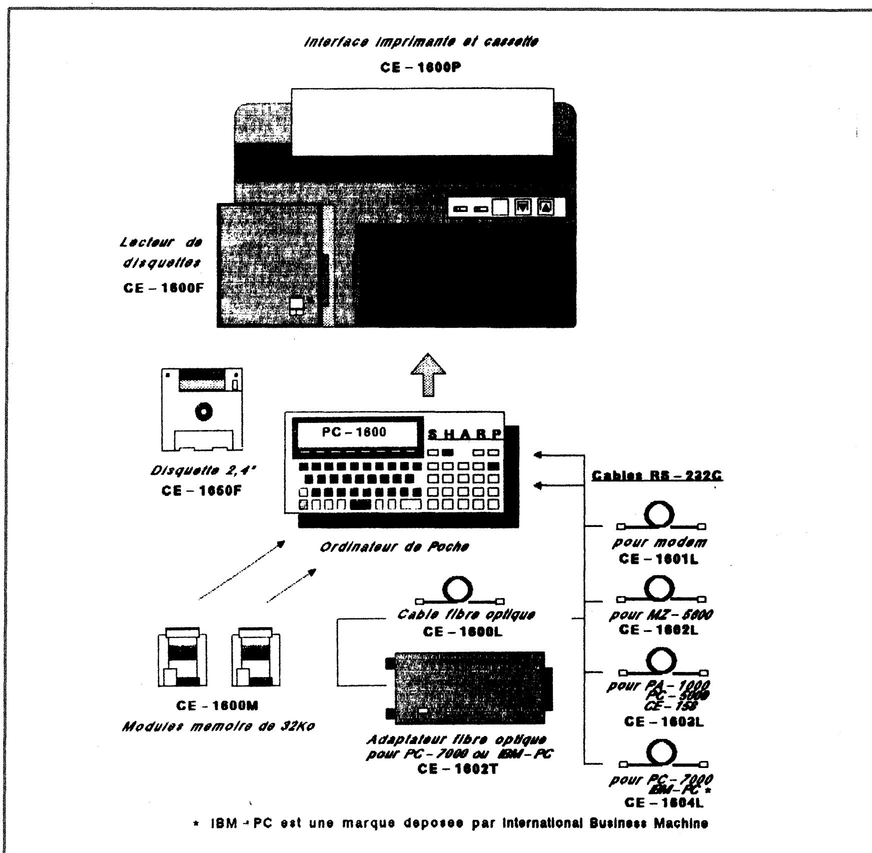
Un jeu complet d'instructions du basic permet au PC 1600 de rester en veille et d'exécuter une tâche à un moment précis programmé par son horloge interne ou commandé par l'interface RS 232. Ces fonctions, très utiles pour la programmation d'envoi ou de reception de messages par modem trouvent également de nombreuses applications dans l'industrie et la recherche : surveillance, contrôles de processus etc.

LA COMPATIBILITE

Digne héritier du PC 1500, le PC 1600 se devait de bénéficier de tous les acquis de son ainé. Là, il faut reconnaître que les concepteurs du PC 1600 ont fait très fort car, tant au niveau logiciel que matériel, la compatibilité entre les deux machines est pratiquement totale. Autour du PC 1600, pourront, en effet, prendre place tous les périphériques du PC 1500 : Interface imprimante CE 150, modules mémoire CE 151, CE 155, CE 159, CE 161, table à digitaliser CE 153, interface parallèle – RS 232 CE 158 etc. Tous les logiciels conçus en basic sur le PC 1500 tourneront également sans difficulté sur le PC 1600. Un second niveau de compatibilité semble également avoir été envisagé puisque le générateur de caractère du « 1600 » est totalement compatible avec celui d'un autre PC, celui d'I.B.M. A ce stade, il nous faudra un peu de patience pour en savoir plus. Ce rapide survol des caractéristiques techniques de notre nouveau PC a été écrit après une toute petite journée de prise en mains, il reste donc beaucoup à découvrir. Pour l'instant, nous sommes encore sur le coup de l'étonnement car, avoir réussi à intégrer sous le même volume qu'un PC 1500, deux microprocesseurs, 176 Ko. de mémoire, une interface série, une interface fibre optique, un afficheur de 4 lignes relève tout de même de l'exploit !

S.B.






Le cable en fibre optique

MULTIPOSTE CONTRE RESEAU LOCAL : UN COMBAT FRATRICIDE

« Qu'est-ce qu'un réseau ? », « Quelles sont les procédures de télécommunications ? », « Qu'est-ce que l'informatique répartie ? » autant de questions posées par ce mot magique et très à la mode dans les salons informatiques : La COMMUNICATION. Si vous posez ces questions aux dits spécialistes, leur réponse passera généralement par 3 phases successives : d'abord un long silence gêné qui vous en apprendra beaucoup sur la complexité du problème posé ; puis un torrent de termes techniques duquel vous pourrez puiser quelques mots dont l'hermétisme et l'ésotérisme vous feront plus penser à un réseau d'espionnage qu'à un réseau de communications : RS 232, V24, CCITT, paquets, jetons... ; enfin viendra la sentence : « Tout système informatique non dédié à la communication est condamné à disparaître à plus ou moins brève échéance ».



Le Système Multipostes SHARP M 56

S'il existe encore une certaine confusion au niveau des moyens, l'avenir de nos micro-ordinateurs, lui semble tout tracé. Tôt ou tard, ces systèmes sont voués à une cohabitation pacifique (tiens, tiens...) et à un dialogue cohérent.

Tout ordinateur trublion désirant faire cavalier seul se verra vite relégué au rang des antiquités dans le placard ou dorment actuellement environ 50% des consoles familiales. En micro-informatique, l'élection, à candidat unique, de **MINITEL** au titre de président à vie a permis la standardisation d'un protocole unique de communications. Là encore, si tout est simple (ou presque) au niveau de l'utilisateur final connecté à sa ligne P.T.T., il en va très différemment du côté des sites centraux, émetteurs des informations et connectés entre eux par un éventail inextricable de réseaux et une foule de procédures de télécommunications.

RESEAUX LOCAUX ET COMMUNICATIONS

Autant il est permis aux hommes de raconter à peu près tout et n'importe quoi, autant on défend catégoriquement à une machine dite intelligente de dire que 2+2=5. Pour bien comprendre le faux problème des télécommunications ou les vraies solutions réseau, il est important de camper le décor et de reprendre quelques notions élémentaires de théorie. Pour cela, prenons notre microscope (un gros, si possible...) et observons... Dans le cœur même de l'ordinateur, entre les mailles atomiques du cristal de silicium, une grande quantité de minuscules particules se déplacent d'un point à un autre, telles des fourmis, au travers de quelques millions de portes dites logiques. Le « Grand Boulevard » du microprocesseur, appelé égalemet **BUS** permet la circulation du flot de ces électrons selon un code de la route bien établi. Ce même code se retrouve dans les différents bus de l'ordinateur où il s'agit de faire la police (arbitrage du bus) et de parer au plus pressé (Gestion des priorités). Sorti de l'enceinte protectrice du boitier de l'ordinateur, le flot d'électrons devient très vulnérable ; si des perturbations se présentent sur son chemin, il ne sera pas étonnant que 2+2 soient égaux à n'importe quoi mais certainement pas à 4. Pour que deux équipements informatiques éloignés de quelques centimètres ou de plusieurs années-lumière puissent communiquer, il leur suffit de deux éléments : un support pour la liaison et un protocole pour s'assurer que l'intégralité des données est bien passée d'une machine vers l'autre. Des premiers ordinateurs mono-tâche, on est rapidement passé aux machines multi-tâches, multi-postes, multi-utilisateurs. Deux obstacles à une bonne transmission restaient à franchir : la distance et les perturbations (parasites).

La distance : dès l'instant ou les nombreux utilisateurs d'un même site central ne pouvaient pas être regroupés dans une même salle ou dans un même bâtiment, les supports de transmissions de données se sont diversifiés, on en trouve maintenant de nombreux : Cable coaxial, ligne téléphonique, fibre optique, satellites etc.

Les pertubations : qui dit distance, dit interruptions intempestives, parasites de toutes sortes, ajout ou suppression d'impulsions, bref, déformations du signal original émis. De plus, ce signal voit son intensité diminuer après avoir parcouru une certaine distance dans le fil qui l'emmène à destination, il est donc nécessaire de le réamplifier régulièrement. Ne parlons pas des lignes publiques P.T.T. dont les « fritures » se mêlent allègrement aux bits qui voyagent. En fait, la migration des octets d'un microprocesseur vers un autre ressemble à un véritable parcours du combattant.

LES TELECOMS

Il a donc été nécessaire d'introduire dans les transferts de données des procédures ou formes de politesse (de l'anglais « is'nt it ») augmentant la fiabilité des échanges d'informations. D'abord assez frustres, puis de plus en plus complexes, on les appelle des protocoles. Utilisée dans un premier temps pour relier un terminal à un site central, la télécommunication est passée d'une liaison point par point à une liaison multi-points grâce au partage de la ligne dans le temps (Multiplexage). Notons enfin une certaine unification, chez certains constructeurs des procédures réseaux et télécoms (ex : le **S.N.A.** d'IBM) ; leur caractéristique étant une relative lenteur malgré le mode multiplexé.

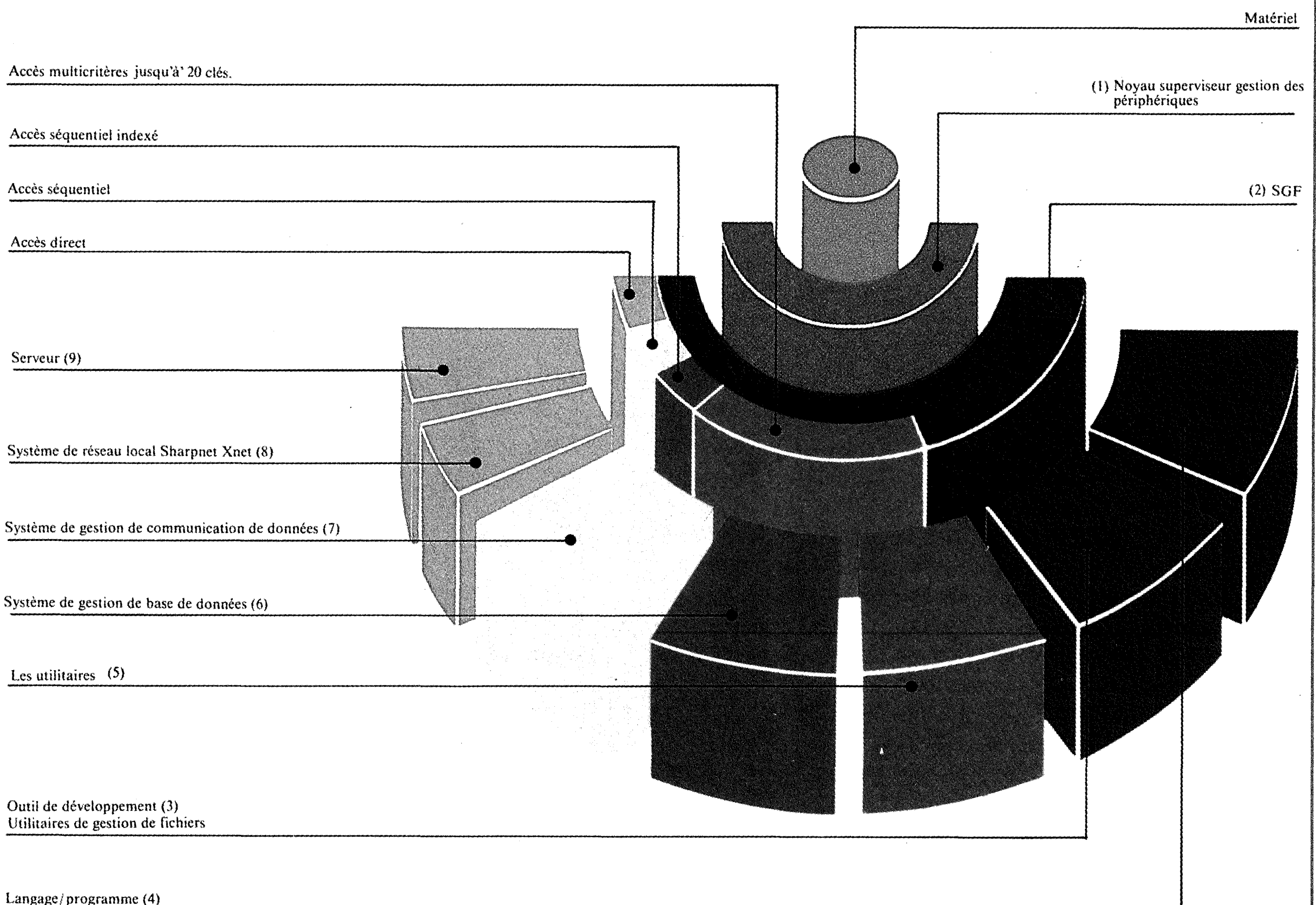
LES RESEAUX

Devant la demande accrue en puissance et capacité réclamée par les utilisateurs, les constructeurs d'ordinateurs ont relié entre eux certaines unités centrales pour optimiser les ressources et augmenter le nombre des postes de travail. Différents protocoles sont utilisés pour assurer la liaison entre ces postes, leurs différences essentielles avec les télécoms résident dans le débit d'informations. En effet, deux ou plusieurs unités centrales doivent communiquer entre elles à très grande vitesse car le volume d'informations à transmettre est énorme. Les réseaux représentent également une forme de stratégie informatique. Face au centralisme des calculateurs géants du numéro 1 mondial (devinez qui...) la réponse consiste à faire travailler en parallèle plusieurs « petits » ordinateurs pour obtenir la puissanxe d'un gros ; cela s'appelle de l'informatique répartie et permet de fractionner un système tant au niveau de l'investissement que de la maintenance.

Les notions de télécom et de réseau sont utilisées depuis très longtemps (c'est très relatif en informatique) sur les systèmes « lourds » conçue à l'origine pour fonctionner en multitâches ; elles se vulgarisent progressivement sur les minis, dans la foulée d'IBM.

Via les télécoms, il est en effet parfaitement possible d'émuler un écran de site central grâce à une carte appropriée et à un logiciel ad-hoc. La liaison s'effectue en mode **TWINAX**

Architecture du Système M 56



(paire torsadée), **COAXSYS** (cable coaxial) ou par une transmission série du type **V24** et un **MODEM** (**MO**dulateur-**DEM**odulateur pour ligne téléphonique). Ces logiciels d'émulation, utilisés principalement sur le PC d'IBM (qu'il faut pas confondre avec un Pocket Computer !) sont actuellement très à la mode pour 2 raisons :

- Le double emploi du PC ; autonome avec sa batterie de logiciels dédiés et en émulation écran-clavier sur un gros système pouvant permettre de lancer simultanément plusieurs applications.
- La mise en commun de fichiers permettant d'utiliser une petite partie du disque du site central comme mémoire de masse pour le micro ; le gros ordinateur étant utilisé comme serveur de fichier en mode local ou en télécom (ex : **TEMPUS, LINK,**...).

Autre utilisation de la communication : le partage des périphérique. Comme il est très onéreux de poster derrière chaque console une imprimante rapide et un disque de grande capacité, ceux-ci sont utilisés en réseau par tous les utilisateurs. Ce partage a été une des premières raisons d'être du réseau local qui devient maintenant un véritable outil de bureau intégrant une messagerie, une connection au télex, au minitel et permettant un dialogue permanent avec des sites centraux, en mode local ou par réseau téléphonique. Le réseau local est également considéré comme une passerelle entre les différentes marques d'ordinateurs et leurs incompatibilités. Il permet d'homogéiniser l'environnement informatique d'une société tout en conservant des machines spécifiques dédiées à des tâches particulières.

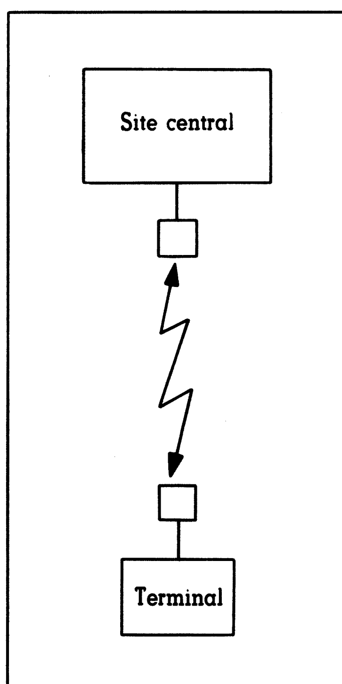
Dans un futur proche, il n'est pas aventureux de prédire une fusion du réseau téléphonique et du réseau local, un mariage des réseaux locaux et de l'image (téléconférence, visiophone).

L'évolution des communications représenterait une épouvantable pagaille si une tentative de normalisation ne venait pas mettre un peu d'ordre dans cette foultitude de connexions.

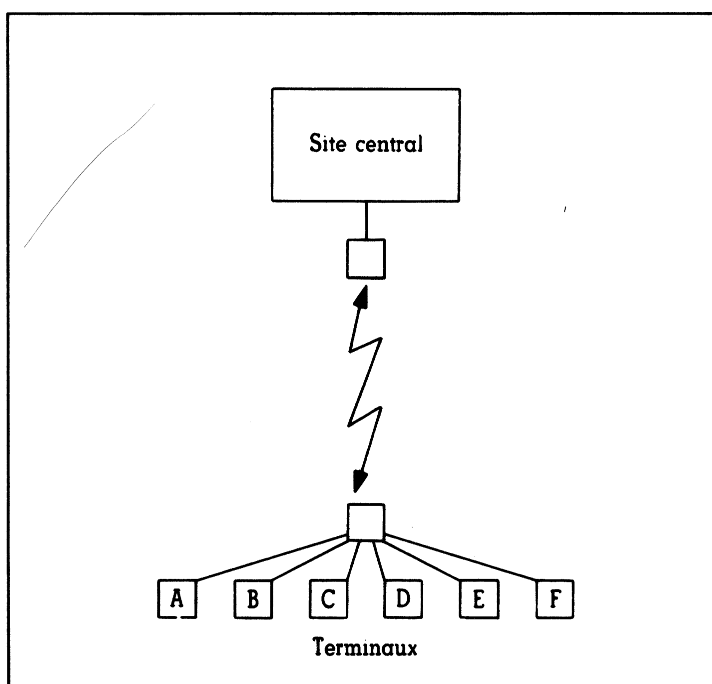
Dans les télécom entre sites centraux, le numéro 1 mondial à imposé des standards de facto adoptés par un certain nombre de constructeurs. Cependant, chaque grand constructeur conserve sa solution propre (BULL, NIXDORF, DEC...) qui est parfois plus évoluée mais qui présente toujours des « passerelles » (solutions Hardware et Software de liaison) avec les protocoles d'IBM. Mêmes constatations pour les réseaux locaux de grands sites ; c'est seulement pour les micros et supermicros qu'ont été adoptés des standards de communications rédigés par l'**OSI**, Organisme de Standardisation de l'Information. Ces normes laissent le choix d'un grand nombre d'interprétations mais gageons que si Big Blue (devinez toujours qui c'est) se positionne sérieusement dans ce domaine comme nous l'avons constaté dans le domaine du standard de l'IBM PC.

La communication, c'est l'outil d'avenir de l'entreprise qui doit, dès maintenant, utiliser ses énormes potentialités pour progresser. Celles qui ne l'auront pas compris sont vouées à l'échec à plus ou moins court terme.

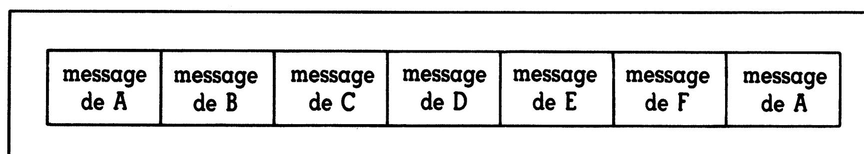
Edouard Bernard
Sylvain Bizoirre



Transmission Point a Point



Transmission Multipoints



Multiplexage : succession de messages

Système d'équations par la méthode JACOBI

Tous PC

Il est facile de résoudre en quelques minutes une équation de ce genre

$2x + y = 1$
$x - y = 2$

La solution étant $x = 1$ et $y = -1$

Il est un peu moins facile de résoudre un système du genre :

$3x - y + z = 8$
$2x + 3y - z = 0$
$x - 5y + 7z = 14$

Les solutions sont : $x = 2$, $y = -1$, $z = 1$.

On les obtient par substitution, élimination ou règle de Crammer.

Imaginons maintenant un problème de ce genre :

$16x_1 + 2x_2 + 3x_3 + x_4 = 15$
$x_1 + 10x_2 + 4x_3 + 3x_4 = 1$
$3x_1 + x_2 + 15x_3 + 2x_4 = 40$
$x_1 + 2x_2 + 4x_3 + 8x_4 = 61$

Il nous faudrait certainement une bonne demi-heure pour le résoudre sans compter les éventuelles erreurs. Un PC nous demandera moins de 30 secondes pour obtenir un résultat...exact !

On doit cette méthode au mathématicien JACOBI. Shématiquement, elle consiste en un processus d'itérations qui, par approximations, s'approche progressivement de la solution réelle.

Supposons que le système d'équation à résoudre soit $Ax = b$. A étant la matrice de coefficient, x l'inconnue (x, x, x,...xn) et b le vecteur de termes indépendants. Nous en déduisons une formule itérative de la forme $X_{r+1} = BX_r + C$. En choisissant une valeur initiale X_0, on peut utiliser ensuite cette formule pour obtenir X_1, X_2, etc.

En tablant sur un processus convergeant (sinon nous entrerions dans des explications trop complexes), les valeurs successives de X_r donneront des approximations de plus en plus proches de la solution réelle X.

EXPLICATIONS DU PROGRAMME

Ligne 65 : Il existe 3 possibilités :

P = 1 : Donne les valeurs successives de x(1), x(2),...X(n) et le nombre d'itérations. On arrête en appuyant sur ON/BREAK quand on pense que l'approximation est suffisante. Possibilité archaique et peu conseillée.

P = 2 : Le calcul se terminera après le nombre d'itérations désirées. (Nombre d'itérations en ligne 68).

P = 3 : Donne la solution finale avec une erreur inférieure à celle demandée en ligne 69. (Option plus scientifique et logique)

Ligne 90 : Possibilité d'imprimer ou non (Dans les cas P = 2 et P = 3, il y a toujours impression).

Lignes 120 à 230 : Introduction :

C(1), C(2),...C(n) = Termes indépendants.

A(I,J) = Matrice de coefficient du système.

La suite du programme calcule le processus itératif selon la méthode de JACOBI.

QUELQUES EXEMPLES

Résoudre :

$8x_1 - x_2 + x_3 = 1$
$2x_1 + 10x_2 - x_3 = 4$
$x_1 + x_2 - 5x_3 = 3$

Solution intiale : x(1) = 0.125
x(2) = 0.4
x(3) = 0.6

Solution avec P = 3 et erreur $< 10^{-5}$

x(1) = 2.249383438E-01
x(2) = 3.056266887E-01
x(3) = 4.938913787E-01

8 Itérations.

Un second exemple plus complique :

$14.8934x_1 + 2.3113x_2 + 3.004\ x_3 + 0.9893x_4 = 13.53285$
$1.0203x_1 + 9.9893x_2 + 4.0177x_3 + 2.9391x_4 = 0.72192$
$2.9090x_1 + 0.8931x_2 + 16.0417x_3 + 1.9393x_4 = -44.0032$
$0.9993x_1 + 2\ x_2 - 3.9\ x_3 + 17.7971x_4 = 60.52743$

En moins de 15 secondes, avec P2 = 2, nous avons la solution initiale :

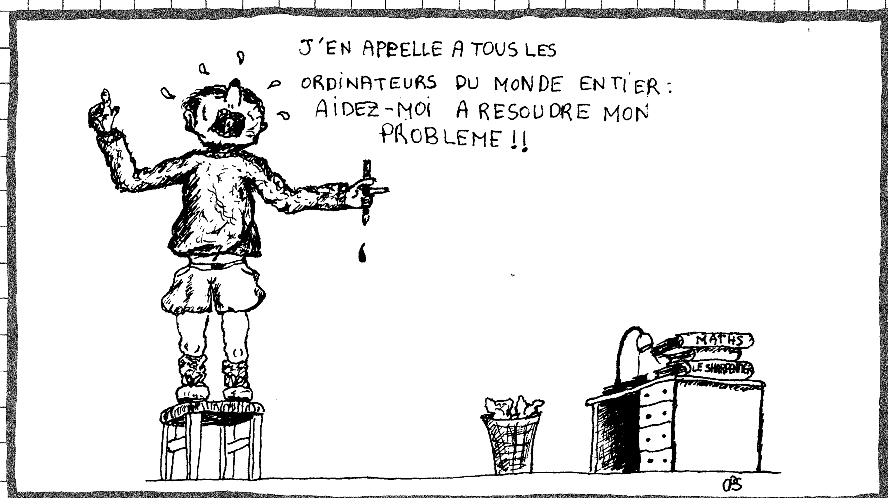
x(1) = 0.9086474546
x(2) = 0.0722693281
x(3) = 2.743050924
x(4) = 3.400971507

Solution finale, après 7 itérations :

x(1) = 1.318864227
x(2) = 0.139515867
x(3) = 3.482427293
x(4) = 4.074445915

Attention cependant : S'il n'y pas de convergence, les solutions trouvées seront de plus en plus éloignées de la réalité.

LUIS MARCHESI ULLASTRES
(Espagne)



```
10:"JAC"REM RESOLUTION_D_UN_SYSTEME_D_EQUATIONS_PAR_LA_ME
   THODE_DE_JACOBI
30:REM COPYRIGHT_PAR_LUIS_IGNACIO_MARCHESI
40:REM *****_METHODE_JACOBI_*****
50:CLEAR :WAIT 0
60:CLS :INPUT "Nb_d_eQuations_=";N
65:INPUT "Choix_No_=";P
68:IF P=2INPUT "Nb_Iterations_=";S
69:IF P=3INPUT "Erreur_<";ER
70:DIM A(N+1,N+1),C(N),X(N),Y(N),P(N),D(N),V(N)
90:INPUT "IMPRESSION_(1/0)_=";IM
120:FOR J=1TO N
130:C$="C("+STR$ J+")="
140:PRINT C$;
150:INPUT C(J):CLS
160:NEXT J
170:FOR I=1TO N
180:FOR J=1TO N
190:A$="A("+STR$ I+","+STR$ J+")="
200:PRINT A$;
210:INPUT A(I,J):CLS
220:NEXT J
230:NEXT I
240:FOR I=1TO N
245:V(I)=0
250:D(I)=C(I)/A(I,I)
260:P(I)=D(I)
270:IF R=0LET X(I)=D(I)
280:NEXT I
290:IF R=0GOTO 400
300:FOR I=1TO N
310:FOR J=1TO N
320:IF I=JGOTO 350
330:Y(I)=A(I,J)*X(J)/A(I,I)
340:P(I)=P(I)-Y(I)
350:NEXT J
360:NEXT I
370:FOR I=1TO N
380:X(I)=P(I)
390:NEXT I
395:IF P=2OR P=3GOTO 470
400:FOR I=1TO N
405:IF I=1BEEP 5,10
410:X$="X("+STR$ I+")="
420:WAIT 120
430:IF P=2OR P=3LPRINT X$;X(I)
435:IF P=2OR P=3AND R>0AND I=NLPRINT R;"__ITERATIONS"
440:IF IM=0AND P=1PRINT X$;X(I);"___";R
445:IF IM=0AND P=1LPRINT X$;X(I);"__";R
450:NEXT I
455:IF P=3AND C=NGOTO 490
460:IF IM=1LF 1
470:R=R+1
475:IF R=SGOTO 400
476:IF R>SAND P=2GOTO 490
478:IF P=3GOSUB 600
479:C=0
480:GOTO 240
490:END
600:FOR I=1TO N
610:FOR J=1TO N
620:V(I)=V(I)+A(I,J)*X(J)
630:IF J=NLET V(I)=V(I)-C(I)
640:IF ABS V(I)<ERLET C=C+1
650:NEXT J
660:NEXT I
670:IF C=NGOTO 400
680:RETURN
```

Bensolver

PC 1350-2500

Problème ancestral classique, la résolution d'équations de degré supérieur à 2 se révèle particulièrement intéressante à réaliser avec les ordinateurs de poche. Nous en établirons d'abord les principes, puis exposerons ensuite différentes méthodes simples pouvant faire l'objet d'un programme.

PRINCIPE

Soit f une fonction numérique et E un sous-ensemble des réels. Résoudre l'équation f(x) = 0 consiste à déterminer les éléments de E, dont l'image par f est nulle. La résolution de telles équations intervient dans de nombreuses branches des mathématiques, ainsi que dans d'autres disciplines scientifiques. Une résolution purement algébrique est parfois possible, et permet de déterminer exactement les solutions de l'équation étudiée. Cette résolution se fait souvent par le biais de changements de variables, qui aboutissent à la résolution d'équations plus simples. Il est cependant clair que ces méthodes ne sont en aucun cas généralisable; nous ne les retiendrons donc pas.
Ainsi, de façon plus générale la résolution de l'équation consiste d'une part, à déterminer des intervalles contenant chacun une unique solution, et d'autre part, à déterminer des valeurs approchées de chaque solution. On s'attache alors à concevoir des algorithmes permettant l'obtention de valeurs approchées successives, avec des incertitudes de plus en plus faibles jusqu'à la précision souhaitée. Non moins intéressante sera la comparaison de leurs performances.

METHODES

Méthode de dichotomie :
Soit f une fonction continue sur un intervalle [a,b], et telle que f(a)*f(b) < 0. L'équation f(x) = 0 admet alors au moins une solution sur l'intervalle [a,b]. En supposant que la solution est unique, on calcule f((a + b)/2). On compare ensuite le signe du résultat avec les signes de f(a) et de f(b). Ce qui permet de situer la solution dans l'un des deux intervalles [a,a + b/2] et [a + b/2,b]. Ce travail est réitéré jusqu'à l'obtention de la précision requise. La dichotomie indexée permet un « balayage » des x possibles, sans contraintes. C'est la méthode utilisée ici.

Méthode d'interpolation linéaire ou des parties proportionnelles :
Dans la dichotomie, on partage systématiquement l'intervalle de séparation de la solution étudiée en deux parties égales. Or dans la plupart des cas, la séparation des racines conduit à étudier la fonction dans un intervalle, où sa courbe est voisine d'une droite. Il semble donc que l'on ait intérêt à utiliser non pas le milieu (a + b)/2, mais l'intersection de l'axe des abcisses (Y = 0) avec la corde (A,B), où A et B sont des points de coordonnées (a,f(a)) et (b,f(b)). Cette intersection aura donc une abcisse c telle que f(a) + (f(b) – f(a))*((c – a)/(b – a)) = 0. Soit c = (af(b) – bf(a))/(f(b) – f(a)). En bref, cette méthode revient à partager l'intervalle [a,b] dans le rapport k = – f(a)/f(b). Le gain en rapidité de convergence allant avec une plus grande complexité. Selon les équations, elle sera plus ou moins rentable.

Méthode de Newton :
Celle-ci est assez différente des précédentes, mais repose sur le même principe. Reprenons les points A(a,f(a)) ct B(b,f(b)) : La méthode de Newton consiste à déterminer les tangentes à la courbe en ces points, et leurs intersections avec l'axe des abcisses. Certains algorithmes

calculent les deux tangentes ; celui que nous vous proposons laisse un des points fixe. Ceci du fait de la relative lenteur des machines de poche. Cette méthode peut être excessivement efficace, mais présente le désavantage de n'être optimum qu'en modifiant les critères de convergence pour chaque fonction. Au risque de ne plus trouver qu'une partie des solutions de chaque équation. De plus les solutions ne sont pas présentées en ordre croissant ou décroissant, et on peut retomber plusieurs fois sur la même racine.

Méthode des sécantes parallèles :
Il s'agit ici uniquement d'une simplification de la méthode de Newton, ci-dessus décrite. En effet, au lieu de considérer les tangentes aux points A et B, on utilise des droites ayant un coefficient directeur correspondant au nombre dérivé f'(a) de la fonction en ce point. Cette méthode permet de gagner en simplicité, mais la perte en vitesse de convergence est très importante : Elle ne fait donc pas l'objet d'un programme. Pour ceux qui voudrait modifier le programme proposé, rappelons que f'(a) = Limite quand x tend vers x0 de (f(x) – f(x0))/(x – x0). Toute suggestion sera naturellement la bienvenue.

REMARQUES ET CONSEILS D'UTILISATION

• Signalons d'abord que les programmes proposés dépassent largement les algorithmes de base, par un souci d'esthétisme et de confort d'utilisation. Vous avez donc la possibilité de les « éplucher » pour économiser la mémoire et augmenter les performances. A vous de voir...

• D'autre part, certaines équations se révèlent parfaitement hérmétiques à ce genre de résolution. Le cas le plus courant concerne les polynôme possédant des racines très rapprochées : Les méthodes employées permettent rarement d'en obtenir la totalité. Une rapide étude de la fonction sera alors plus efficace...

Exemple :
100x 3 – 22x 2 – x + 0.22 = 0 s'écrit :
100(x – 0.1) (x + 0.1) (x – 0.22) (résolution algébrique)
Newton donne x1 = 0.22, x2 = 0.1 en 60 s. et l'interpolation donne x1 = – 0.1 et x2 = 0.22 en 85 s., tandis que la dichotomie ne donne rien !!

• Mention spéciale pour les équations trigonométriques admettant souvent une infinité de solutions du type x = PI/k + k★PI. Leurs résolutions sont très possibles et rapides. Mais méfiez vous de la méthode de Newton qui peut amener à de fausses généralisations. Utilisez surtout la dichotomie.

Eemples :
La fonction COS (x – PI)★ SIN (x + PI) admet 13 solutions de – 10 à 10 s'écrivant x = PI/2 + k★PI et x = PI + k★PI. La dichotomie les trouve toutes en moins de 5mn à 1E – 8 près, Newton met 40 s. pour en trouver d'abord 3 et n'en trouvera finalement que 7 entre – 10 et 10.

• De façon générale, la méthode de Newton est parfaite pour des polynômes de degré < 5, au-delà on choisira l'interpolation linéaire et la dichotomie pour les équations trigonométriques.

• Il existe bien sur d'autres méthodes dont celle de Lagrange, celle de Ferrari et la célèbre de Bairlow (sol. complexe).

UTILISATION DE BENSOLVER

Le programme occupe 3 Ko et fonctionne tel quel sur un PC 1350 et 2500 (voir plus loin). Tous les branchements se font grâce à des labels (étiquettes), les numéros de ligne sont donc purement indicatifs et peuvent être modifiés à loisir. On démarre par RUN ou DEF A. Il faut alors opter pour l'impression ou non des résultats. Puis on définit la précision des résultats, qui joue bien sûr sur le temps de recherche. Mais si une précision de 1E – 8 est parfaite, les valeurs 1E – 1 ou 1E – 2 n'ont quasiment aucun sens.
Ensuite, on choisit dans le menu proposé une des trois méthodes décrites, en fonction de la fonction entrée au préalable en MODE PRO, à la ligne 250. L'impression du titre et de la méthode choisie, ainsi que de l'équation s'opère alors. Le clignotement de la flèche (↦) en haut à gauche de l'écran, indique le bon déroulement de la recherche. Qui dure de 25 s. pour des solutions entières, à plusieurs minutes pour des équations plus complexes. On remarquera que la recherche s'effectue de – 10 à 10 pour la dichotomie et l'interpolation, alors que la méthode de Newton utilise un intervalle sensible plus étendu : L'ensemble des réels. Ce qui amène parfois des racines très étonnantes !!

Les programmes s'intérrompent dans les cas suivants :
1) En cas d'erreur due à l'ensemble de définition de la fonction (LN 0 par ex.)
2) Lorsque l'écran du PC est plein, c'est-à-dire lorsqu'il y a 3 racines d'affichées. Un appui sur ENTER est alors nécessaire. (Cette mesure de sécurité s'est révélée indispensable).
3) Lorsque durant la recherche a lieu un appui prolongé sur une quelconque touche alphanumérique.
4) Pour la dichotomie et l'interpolation, lorsque l'intervalle (– 10, 10) a été entièrement examiné. Un appui sur ENTER ramène alors au début.
5) Pour la méthode de Newton, quand plus de 30 grandes boucles ont été faites, soit à peu près 10 minutes de recherche.
D'autre part, si une équation se révèle insoluble, on peut modifier les critères suivants : pour la méthode de Newton en ligne 380, les variables A et W déterminent l'approche de la courbe. (essayez 1E – 2 ou 1E – 4), et le pas de recherche (essayez 1/9, 2/3, 3/4, ou 5/6). Pour les deux autres, on modifiera en ligne 560 les variables R et W. Celà n'étant que très rarement nécessaire.
Les possesseurs de PC 2500 ne devront pas oublier de se placer en mode majuscule (CAPS), et devront modifier la ligne 290 afin de remplacer les valeurs &6F22 et &6F23, respectivement par &6DB2 et &6DB3 (DATA PTR). Pour les fonctions trigonométriques se placer dans le mode angulaire désiré.

Benoit CHAFFANJON

DESCRIPTION DU PROGRAMME

200-240 : Commentaires
250 : Ecriture de la fonction choisie
270-280 : Fin de la routine de la fonction
290-310 : Présentation et branchements sur « + » et « / »
320-370 : Choix de la méthode de résolution
380 : Début de la méthode de Newton. Init. des variables
390-430 : Algorithme et boucle de recherche
450 : Détermine l'arrêt de la recherche
460-470 : Boucle pour éviter les répétitions de racines
480 : Ecriture et impresseion des résultats
490-520 : Fin de cette méthode
530-540 : Début de l'interpolation
550 : Début de la dichotomie
560 : Initialisation des variables
570-700 : Boucle de recherche commune
710 : X suivant pour l'interpolation
720 : X suivant, dichotomie
725-740 : Eventuel arrêt recherche et contrôle racines

750 : Ecriture et impression des solutions
760-820 : Fin des deux algorithmes
830 930-990 : Impression des commentaires et résultats
880-920 : Présentation et questions de début de programme
1000-1120 : Obtention de la fonction stockée dans FX($(0)
1130-1150 : Fin du programme

```
190:GOTO "A
200:"-------------------
210:"(c)FEVRIER-1986-BC-
220:"-------------------
230:"-----BENSOLVER-----
240:"-------------------
250:">"Y=X^4-2*X^3+X^2-X-1
270:"-------------------
280:RETURN
290:"A" WAIT 0: CLEAR : CLS : DIM FX$(0
    )*80: RESTORE ">":DE= PEEK &6F22+
    PEEK &6F23*256
300:CURSOR 28: PRINT "Benoit Chaffanjon
    ": CURSOR 56: PRINT "Presente":
    GOSUB "+
310:CURSOR 25: PRINT "Resolution d^equa
    tions": CURSOR 53: PRINT "dans les
    reels": GOSUB "+": GOSUB "/"
320:CLS : PRINT "1)> Dichotomie indexee
    ..": PRINT "2)> Methode de Newton..
    ."
330:PRINT "3)> Interpolation.......":
    PRINT "4)> Fin.................."
340:"*" ON VAL INKEY$ GOTO "w","z","y",
    ":
350:GOTO "*
360:":" CLS : END
370:"-------------------
380:"z" CLS : WAIT 0:V=0,A=1E-3,W=1/2,G
    =V,F=V,Z=0: DIM Q(9): IF SI$="O"
    LET ZX=1: GOSUB "Z"
390:"1" IF Z=0 GOTO "2
400:G=G-W,C=G: GOTO "3
410:"2"C=G,Z=1: CLS : CURSOR 7: PRINT "
    Methode de Newton"
420:"3" GOSUB "5":F=F+W,C=F: GOSUB "5":
    IF INKEY$ <>"" THEN "A
425:GOTO "1
430:"5" GOSUB "s":X=C: GOSUB ">":B=Y,X=
    A+C: GOSUB ">":D=C,C=D-A*B/(Y-B):
    IF ABS (D-C)>T GOTO "5
440:IF Q=0 GOSUB "9
450:BT=BT+1: IF BT>30 BEEP 1: WAIT :
    PRINT : GOTO "A
460:FOR I=0 TO Q-1: IF ABS (C-Q(I))<1E-
    5 LET I=Q,UU=7
470:NEXT I: IF UU=7 LET UU=0: RETURN
480:"9"Q(Q)=C,X=C,P=0,Q=Q+1:UY=UY+1:
    CURSOR (UY*24): PRINT ">>X"; STR$ Q
    ;"=";C;"<<": IF SI$="O" GOSUB "(
490:IF UY<>3 RETURN
500:UY=0: BEEP 1: WAIT : PRINT : WAIT 0
    : RETURN
510:"6" WAIT : PRINT : GOTO "A
520:"-------------------
530:"y"IL=9: CLS : CURSOR 9: PRINT "Int
    erpolation": IF SI$="O" LET ZX=3:
    GOSUB "Z
540:GOTO "q
550:"w" CLS : CURSOR 13: PRINT "Dichoto
    mie": IF SI$="O" LET ZX=2: GOSUB "Z
    "
560:"q"R=3/4,W=1/2,A=-10,B=A+1
570:"h"X=A: GOSUB ">":C=Y,X=B: GOSUB ">
    ": GOSUB "s": IF ABS C<T THEN "b
580:IF ABS Y<T THEN "c
590:GOTO "d
600:"b"X=A: GOTO "e
610:"c"X=B: GOTO "e
620:"d" IF C<0 AND Y>0 GOTO "f
630:IF C>0 AND Y<0 GOTO "g
640:A=A+W,B=B+W: IF X<10 GOTO "h
650:BEEP 1: WAIT : PRINT : GOTO "A
660:"f"D=A,E=B: GOTO "i
670:"g"D=B,E=A
680:"i" IF IL<>9 GOTO "o
690:IF E>D LET GG=E,E=D,D=GG
700:X=D: GOSUB ">":C=Y:X=E: GOSUB ">":
    GOSUB "s
710:X=(D*Y-E*C)/(Y-C): GOTO "p
720:"o"X=(D+E)/2: GOSUB "s
725:"p" IF INKEY$ <>"" THEN "A
730:GOSUB ">": IF ABS Y>T THEN "!
740:"e" IF ABS (X-M)<.1 THEN "j
750:Q=Q+1:UY=UY+1: CURSOR (24*UY):
    PRINT ">>X"; STR$ Q;"=";X;"<<":H=H+
    1,M=X: IF SI$="O" GOSUB "(
760:IF UY<>3 THEN "j
770:UY=0: BEEP 1: WAIT : PRINT : WAIT 0
780:"j"A=A+R,B=B+R: GOTO "h
790:"!" IF Y>0 THEN "m
800:D=X: GOTO "i
810:"m"E=X: GOTO "i
820:"-------------------
830:"(" LPRINT ">>X"; STR$ Q;"=";X;"<<"
    : LPRINT "------------------------"
    : RETURN
840:"s" IF WQ=1 THEN "t
850:CURSOR 0: PRINT "-->":WQ=1: RETURN
860:"t" CURSOR 0: PRINT "   ":WQ=0:
    RETURN
870:"+" LINE (2,2)-(147,29),B: WAIT 120
    : GPRINT : WAIT 0: CLS : RETURN
880:"/" CLS : CURSOR 24: PRINT "Sortie
    sur imprimante ?": CURSOR 56: PRINT
    "(O/N)"
890:"u"SI$= INKEY$ : IF SI$="" THEN "u
900:CLS : CURSOR 25: PRINT "Precision s
    ouhaitee :": CURSOR 56: PRINT "1E-"
    : WAIT 15
910:","T= VAL INKEY$ : CURSOR 59: PRINT
    "?": CURSOR 59: PRINT " ": IF T=0
    THEN ",
920:T=10^-T: WAIT 0: RETURN
930:"Z": LPRINT "======================
    ==": LPRINT "  RESOLUTION D^EQUATIO
    N"
940:"Y" LPRINT "=======================
    =": ON ZX GOSUB "W","E","R
950:GOTO "T
960:"W" LPRINT "     Methode de Newton":
    RETURN
970:"E" LPRINT "    Dichotomie indexee":
    RETURN
980:"R" LPRINT " Interpolation lineaire
    ": RETURN
990:"T" LPRINT "=======================
    =":CL=0
1000:FOR I=DE TO DE+500: IF PEEK I=62
     AND PEEK (I+2)=89 LET CL=I+4,I=DE
     +501
1010:NEXT I: IF CL=0 RETURN
1030:FOR I=CL TO CL+80:TR= PEEK I: IF
     TR<&91 THEN "U
1040:TP=TR-&91: ON TP GOSUB "Z","X","C
     ","V","B
1045:FX$(0)=FX$(0)+KL$: GOTO "I
1050:"Z"KL$="LOG": RETURN
1060:"X"KL$="EXP": RETURN
1070:"C"KL$="√": RETURN
1080:"V"KL$="SIN": RETURN
1090:"B"KL$="COS": RETURN
1100:"U"FX$(0)=FX$(0)+ CHR$ TR
1110:"I" IF PEEK (I+1)=&0D LET I=CL+81
1120:NEXT I:FX$(0)=FX$(0)+"= 0":
     LPRINT FX$(0): LPRINT "==========
     ==============": RETURN
1130:"-------------------
1140:"(c)-Ben.Chaffanjon
1150:"-------------------
```

```
========================
 RESOLUTION D^EQUATION
========================
   Dichotomie indexee
========================
COSX*SINX= 0
========================
>>X1=-9.424777957<<
------------------------
>>X2=-7.853981632<<
------------------------
>>X3=-6.283185306<<
------------------------
>>X4=-4.71238898<<
------------------------
>>X5=-3.141592654<<
------------------------
>>X6=-1.570796329<<
------------------------
>>X7=0.<<
------------------------
>>X8=1.570796329<<
------------------------
>>X9=3.141592654<<
------------------------
>>X10=4.71238898<<
------------------------
>>X11=6.283185306<<
------------------------
>>X12=7.853981632<<
------------------------
>>X13=9.424777957<<
------------------------
>>X14=10.99557428<<
------------------------
```

```
========================
 RESOLUTION D^EQUATION
========================
   Methode de Newton
========================
X^4-2*X^3+X^2-X-1= 0
========================
>>X1=-4.840283022E-01<<
------------------------
>>X2=1.897179401<<
------------------------
```

```
========================
 RESOLUTION D^EQUATION
========================
 Interpolation lineaire
========================
X^4-2*X^3+X^2-X-1= 0
========================
>>X1=-4.840283003E-01<<
------------------------
>>X2=1.897179401<<
------------------------
```

Logique Math

Tous PC

Ce programme purement Basic est écrit sur un PC 1450, mais est en fait compatible avec tous les PCs. Il occupe 1.3Ko et moins de 2.5 lors de l'utilisation. Il s'agit en clair d'analyser les fonctions logiques utilisant les connecteurs usuels de base, et d'établir la table de vérité.

Les connecteurs utilisables sont :

« V » pour OU ou OR

« & » pour ET ou AND

« => » pour IMPLIQUE

« <=> » pour EQUIVALENT

Le programme se lance par RUN ou DEF L. Entrer ensuite une fonction logique dont les variables sont constituées chacune d'une lettre majuscule de l'alphabet parmi 25 (moins V). Cette formule doit comprendre les seuls connecteurs énumérés ci-avant, et les fonctions logiques doivent être séparées par des parenthèses. Voir les exemples. Un BEEP sonore annonce l'affichage d'un résultat. L'impression s'obtient en tapant la commande PRINT =LPRINT avant le DEF L. La formule peut comprendre jusqu'a 80 caractères avec 16 variables au maximum (en théorie). C'est-à-dire 65536 cas possibles, pas mal n'est-ce pas ? Le message d'erreur « Formule Illegale » s'affiche en cas d'oubli de parenthèses (fermées ou ouvertes), ou en cas de présence de caractères non reconnus. Une mauvaise syntaxe dans l'écriture de la formule est aussi détectée. Lisez aussi l'article sur le sujet, de M.G., cela ne pourra que vous êtres profitable.

Thierry ALLARY.

```
Formule:
(A=>(B=>C))<=>((A&B)=>C)

Variables:
ABC

000 -> 1
001 -> 1
010 -> 1
011 -> 1
100 -> 1
101 -> 1
110 -> 1
111 -> 1


Formule:
((A&B)=>(CVD))<=>((C=>D)V(B=>D))

Variables:
ABCD

0000 -> 1
0001 -> 1
0010 -> 1
0011 -> 1
0100 -> 1
0101 -> 1
0110 -> 0
0111 -> 1
1000 -> 1
1001 -> 1
1010 -> 1
1011 -> 1
1100 -> 0
1101 -> 1
1110 -> 0
1111 -> 1
```

```
  1:"(c) T. ALLARY
 10:"L" CLEAR
 20:DIM H$(0)*80,E$(0)*80
 25:PAUSE "**LOGIQUE MATH**"
 30:INPUT "Formule:";E$(0)
 40:E$(0)="("+E$(0)+")"
 50:"VARIABLES
 60:FOR I=1 TO LEN E$(0)
 70:C$=MID$ (E$(0),I,1)
 80:IF C$="V" GOTO 100
 90:IF C$<CHR$ 91 AND C$>"@" GOSUB 120
100:NEXT I
110:GOTO 180
120:FOR J=1 TO T
130:IF C$=MID$ (VA$,J,1) GOTO 170
140:NEXT J
150:T=T+1
160:VA$=VA$+C$
170:RETURN
180:BEEP 1: PAUSE "Variables:": WAIT 70
    : PRINT VA$: PRINT "": WAIT
190:"PROGRAMME
200:LV=LEN VA$:LF=LEN E$(0)
210:FOR I=0 TO 2^LV-1
220:GOSUB 270
230:GOSUB 370
240:GOSUB 450
250:NEXT I
260:GOTO 10
270:"BINAIRE
280:X=I:BI$=""
290:IF X<2 LET BI$=STR$ X+BI$: GOTO 350
300:Q= INT (X/2)
310:R=X-(2*Q)
320:BI$=STR$ R+BI$
330:X=Q
340:GOTO 290
350:IF LEN BI$<LV LET BI$="0"+BI$: GOTO
    350
360:RETURN
370:"INSERT BIN
380:H$(0)=E$(0)
390:FOR M=1 TO LF
400:FOR N=1 TO LV
410:IF MID$ (VA$,N,1)=MID$ (H$(0),M,1)
    LET H$(0)=LEFT$ (H$(0),M-1)+MID$ (B
    I$,N,1)+RIGHT$ (H$(0),LF-M)
420:NEXT N
430:NEXT M
440:RETURN
450:"POSITION
460:T=0
470:FOR J=1 TO LEN H$(0)
480:C$=MID$ (H$(0),J,1)
490:IF C$="(" LET O=J:T=T+1
500:IF C$=")" GOSUB 530
510:NEXT J
520:GOTO 570
530:F=J:T=T-1
540:J=LEN H$(0)
550:IF T<0 GOTO 680
560:RETURN
570:"VALEUR
580:LG=F-O-3:S=O+2
585:IF LG<0 OR LG>3 GOTO 680
590:IF MID$ (H$(0),O+1,1)="-" LET X=VAL
    (MID$ (H$(0),O+2,1)):A=1-X:O=O+1:F=
    O+1: GOTO 700
600:IF MID$ (H$(0),F-2,1)="-" LET Y=VAL
    (MID$ (H$(0),F-1,1)):A=1-Y:F=F-1:O=
    F-1: GOTO 700
610:X=VAL (MID$ (H$(0),O+1,1))
620:Y=VAL (MID$ (H$(0),F-1,1))
630:C$=MID$ (H$(0),S,LG)
640:IF C$="&" LET A=X*Y: GOTO 690
650:IF C$="V" LET A=X+Y-X*Y: GOTO 690
660:IF C$="=>" LET A=1-X+X*Y: GOTO 690
670:IF C$="<=>" LET A=1-X-Y+2*X*Y: GOTO
    690
680:BEEP 2: PRINT "Formule Illegale":
    GOTO 10
690:IF O=1 BEEP 1: PRINT BI$;" -> ";
    STR$ A: RETURN
700:H$(0)=LEFT$ (H$(0),O-1)+STR$ (A)+
    RIGHT$ (H$(0),LEN H$(0)-F)
710:IF LEN H$(0)=3 BEEP 1: PRINT BI$;"
    -> ";STR$ A: RETURN
720:GOTO 450
```

SHARP ENSEIGNEMENT

Pivot de GAUSS

Tous PC

Voici enfin le programme que tous les Sharpentiers attendaient : Une émulation du H.P 15C en ce qui concerne l'aspect calcul matriciel. En effet, le programme qui suit permet, à partir d'une matrice A N X N (N⩽B, voire après quelques modifications N⩽11)
• d'inverser cette matrice (si elle est inversible, bien sûr !)
• de calculer alors le déterminant
• de transformer la matrice initiale en une matrice équivalente triangulaire (donc sans changer son rang)
• de déterminer le rang de A
• de résoudre tous systèmes linéaires réels, admettant A comme système homogène.
• de déterminer les vecteurs propres associés à une valeur propre
• enfin de multiplier deux matrices N * N (N⩽8) (Tout ceci étant contenu dans 1,4 Ko environ).
Examinons le programme :

ENTREE DES DONNEES

Un Run (impératif) lance le programme. « N = » s'affiche, attendant comme réponse la dimension de votre matrice A.
L'introduction de la matrice initiale A a alors lieu. Après un bref A_{ij}, vous devez entrer le coefficient A_{ij}. (et ainsi de suite de A_{11}, A_{12}... à A_{NN}).
En cas d'erreur, notez la position du coefficient, continuez l'introduction des coefficients jusqu'au message : « Que faites-vous ? ». Une pression sur DEF V vous permettra de corriger le coefficient erroné : Vous introduisez i, puis j, puis la vraie valeur et ceci autant de fois que cela s'avère nécessaire.

INVERSE, TRIANGULARISATION, DETERMINANT

Une simple pression sur ENTER, après le message « Que faites-vous ? », va déclencher ce calcul (qui dure environ, pour N=4, 15 secondes).
« DET A= » s'affiche alore.
Une pression sur ENTER donne $C=A^{-1}$.
Attention, afin d'éviter les nombres décimaux peu explicites (saviez vous que 0,142857 valait...1/7), C apparaît sous la forme

$$C=\frac{1}{DET\ A}(C_{ij}*DETA)$$

N'oubliez donc pas de multiplier par $\frac{1}{DET\ A}$
D'ailleurs, la ligne 212 se charge de vous le rappeler). Une remarque importante : La précision de votre calculateur peut parfois introduire des –, 99...99. Dans ce cas vous lirez, par exemple, 16 au lieu de 15,99. Si vous continuez à appuyer sur ENTER, la matrice A apparaît sous sa forme triangulaire. Si A n'est pas inversible, le message « Non inversible » apparaît.

CALCUL DU RANG

• DEF L lance ce calcul.
• Si A est inversible rang A=n et donc « KER A=(0) ».
• Si A n'est pas inversible, après quelques secondes apparaît « RANG A= » puis des vecteurs de base de KER A. Ceux-ci sont donc des vecteurs propres associés à la valeur 0. Si vous vouliez calculer les vecteurs propres assiciés à la valeur, n'introduisez pas A mais « A – ID ».

RESOLUTION DE SYSTEME LINEAIRE A Y=X

• Introduisez dans la machine la matrice A (opération 1))
• Puis la matrice X grâce à DF X (vous introduirez X1, puis X_n).
• Calculez d'abord A^{-1} (grâce à 1 pression sur ENTER après « Que faites-vous »). Puis DEF S,
• Les solutions Y_1, Y_2,... Y_n apparaissent au fur et à mesure.
• Pour les revoir, presser DEF S.

PRODUIT DE MATRICE : PRODUIT A.C. (QUI EST DIFFERENT DE C.A.)

• Introduisez d'abord A (opération 1))
• Puis C en appuyant sur DEF C.
• La démarche est la même que pour A.
• Pour corriger les erreurs de C, appuyez sur DEF Z
• Appuyez alors sur DEF M.
• Le produit A.C. est transféré dans la matrice C.
• L'affichage est le même que pour $C=A^{-1}$, mais ici le DET A=n'a plus de sens et ne doit être pris que comme variable égale à l'unité.
L O G I Q U E 2

CONCLUSION

Il se peut que vos matrices A et A^{-1} aient un déterminant et des coefficients supérieurs ou égaux à 1000.

Un message d'ERROR apparaîtra alors. Il vous suffit en fait de modifier l'instruction USING à la ligne 200 en rajoutant des #.

Attention, la matrice triangulaire obtenue n'est pas la matrice triangulaire « semblable » à A... ; ce n'est donc pas un calcul de trigonalisation.

Ce programme est constitué de nombreuses boucles, ce qui rend son exécution lente. Mais dans une certaine mesure, le calcule de A^{-1} se fait plus rapidement que sur un H.P 15C ; en effet, sur ce dernier, vous obtenez soit A^{-1} soit DET A (et une fois DET A obtenue, A^{-1} n'existe plus, du moins n'existe plus que sous sa forme « LU », alors que votre PC vous donnera **en même temps** A^{-1}, DET A et une matrice triangulaire **équivalente**.

Ceci vaut bien les quelques secondes perdues ! Qui plus est, en supprimant toutes les lignes entre 250 et 396, vous pourrez inverser des matrices 10 X 10 (!) -(ce que ne fait pas un H.P 15 C)...

H. SZYCHTER

```
1:"PIVOT DE GAUSS
3:"G" CLEAR :INPUT "N=
   ";N:DIM A(N,N),C(N,
   N),X(N)
5:WAIT 40:FOR I=1 TO N
   :FOR J=1 TO N:PRINT
   "A "+STR$ I+STR$ J:
   INPUT A(I,J):NEXT J:
   C(I,I)=1:NEXT I
25:WAIT :DP=1:PRINT "QU
   E FAITES VOUS ?"
45:"D"TR=0:RG=N:DP=1
50:FOR F=1 TO N-1:FOR I
   =F TO N
60:IF ABS A(I,F)<>0 LET
   I0=I:GOTO 100
70:NEXT I:NEXT F:RG=I0
71:IF A(N-1,N)<>0 LET R
   G=N-1
72:PRINT "NON INVERSIBL
   E":GOTO 520
100:B=A(I0,F):FOR I=F TO
   N:IF I=I0 OR I=N+1
   GOTO 120
110:C=A(I,F)/B:FOR J=1
   TO N:A(I,J)=A(I,J)-A
   (I0,J)*C:C(I,J)=C(I,
   J)-C(I0,J)*C:NEXT J
120:NEXT I:IF I0=F GOTO
   130
121:FOR J=1 TO N
125:D=A(F,J):A(F,J)=A(I0
   ,J):A(I0,J)=D
127:D=C(F,J):C(F,J)=C(I0
   ,J):C(I0,J)=D:NEXT J
   :TR=TR+1
130:NEXT F:IF ABS A(N,N)
   <1E-7 LET RG=N-1:
   PRINT "NON INVERSIBL
   E":GOTO 515
150:FOR F=N TO 1 STEP -1
   :B=A(F,F):FOR I=F-1
   TO 1 STEP -1:FOR J=1
   TO N:C(I,J)=C(I,J)-C
   (F,J)*A(I,F)/B
155:NEXT J:NEXT I:FOR J=
   1 TO N:C(F,J)=C(F,J)
   /B:NEXT J:DP=DP*B:
   NEXT F:DP=DP*(-1)^TR
200:PRINT "DET= "; USING
   "####.##";DP
210:WAIT :FOR I=1 TO N:
   FOR J=1 TO N:PRINT "
   C "+STR$ I+" "+STR$
   J+":";C(I,J)*DP:NEXT
   J:NEXT I
212:PRINT "ATTENTION AU
   DET":PRINT "*RCP ";D
   P:PRINT "*1/";DP
220:FOR I=1 TO N:FOR J=1
   TO N:PRINT "A "+STR$
   I+" "+STR$ J+":";A(I
   ,J):NEXT J:NEXT I:
   END
250:"L" IF RG=4 PRINT "K
   ER=<0>":END
255:FOR F=RG TO 1 STEP -
   1:B=A(F,F):IF B=0
   GOTO 265
256:FOR I=F-1 TO 1 STEP
   -1:C=A(I,F)/B:FOR J=
   1 TO N:A(I,J)=A(I,J)
   -A(F,J)*C:NEXT J
257:NEXT I:FOR J=1 TO N:
   A(F,J)=A(F,J)/B:NEXT
   J
265:NEXT F:PRINT "RANG A
   =";RG:FOR K=1 TO N-R
   G:PRINT "VECTEUR N0
   ";STR$ K:A$="U ":
   USING "####.##":FOR
   J=1 TO N
267:IF J=N+1-K LET C=1:
   GOTO 270
268:C=-A(J,N+1-K)
270:PRINT "U ";STR$ K;"
   ";STR$ J;":";C:NEXT
   J:NEXT K:END
300:"S" FOR I=1 TO N:B=0
   :FOR J=1 TO N:B=B+C(
   I,J)*X(J):NEXT J:
   PRINT B:NEXT I:END
350:"M" FOR F=1 TO N:FOR
   I=1 TO N:B=0:FOR J=1
   TO N:B=B+C(F,J)*A(J,
   I):NEXT J:X(I)=B
355:NEXT I:FOR I=1 TO N:
   C(F,I)=X(I):NEXT I:
   NEXT F:GOTO 210
370:"C" WAIT 40:FOR I=1
   TO N:FOR J=1 TO N:
   PRINT "C "+STR$ I+
   STR$ J:INPUT A(I,J):
   NEXT J:NEXT I:GOTO 2
   5
380:"X" WAIT 40:FOR I=1
   TO N:PRINT "X ";I:
   INPUT X(I):NEXT I:
   GOTO 25
385:"Z" INPUT "I,J,X";I,
   J,B:C(I,J)=B:GOTO 25
390:"V" INPUT "I,J,X";I,
   J,B:A(I,J)=B:GOTO 25
```

Quadrature Gaussienne à 16 points

Basic

Ce programme effectue des approximations d'intégrales sur des intervalles finis par la méthode des quadratures à seize points de GAUSS-LEGENDRE.

$\int_a^b F(x)\,dx$ est alors calculée en seize courtes itérations.

Le résultat est presque immédiat.

Mettre en ligne 5 000 la fonction sous la forme de celle présentée dans le listing.

DEF A pour lancer le programme.


```
2004:"α"CLEAR :DIM W(16),Z(16):RESTORE "α":RADIAN
2005:DATA .1894506105,.182603415,.1691565194,.1495959888,.1
     246289713,.09515851168
2006:DATA .06225352394,.02715245941
2007:DATA .09501250984,.2816035508,.4580167777,.6178762444,
     .7554044084,.8656312024
2008:DATA .9445750231,.989400935
2010:FOR I=1TO 16STEP 2:READ W(I):W(I+1)=W(I):NEXT I
2011:FOR I=1TO 16STEP 2:READ Z(I):Z(I+1)=Z(I):NEXT I
2012:RETURN
2110:"A"GOSUB "α":PAUSE "INTEGRATE f(x) from A to B":INPUT "A=";A,"B=";B:S=0
2120:FOR I=1TO 16:X=(Z(I)*(B-A)+B+A)/2:GOSUB "fo"
2125:S=S+(Y*W(I)):NEXT I
2130:S=S*(B-A)/2:PRINT S:END
5000:"fo"Y=X
5999:RETURN
```

Fonctions logiques

PC 1500

Ce sont des fonctions fondamentales qu'il n'est pas obligatoire d'avoir comprises pour programmer en BASIC, mais qui permettent toutefois de simplifier de nombreux problèmes.

ELECTRONIQUE

La plupart des composants essentiels d'un ordinateur sont crées grâce à une combinaison de fonctions logiques. Dans ces composants circulent deux courants de tension différente, l'un étant de voltage plus fort que l'autre. Celui de voltage plus fort que l'autre sera le niveau haut = 1 et l'autre le bas = 0. Ils peuvent être de 5V-0V ou 9V-0V selon la technologie de ces composants. Il y a 7 fonctions courantes utilisées.
Soit 2 courants arrivant dans un de ces composants :

OR

	0	1
0	0	1
1	1	1

AND

	0	1
0	0	0
1	0	1

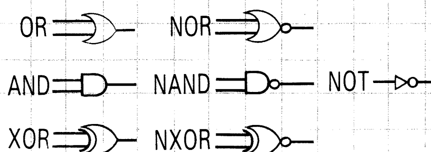
XOR ou EOR

	0	1
0	0	1
1	1	0

Pour mémoriser ces tables :
OR = ou = l'un ou l'autre ou les deux
AND = et = l'un et l'autre
XOR = ou exclusif = l'un ou l'autre mais pas les deux
NOT est une fonction qui ne demande qu'un courant, et elle l'inverse :
NOT 0 = 1
NOT 1 = 0
Remarquez que le NOT c'est égal à 1 XOR c.
Il existe aussi les fonctions NOR, NAND et NXOR qui correspondent respectivement à NOT OR, NOT AND et NOT XOR

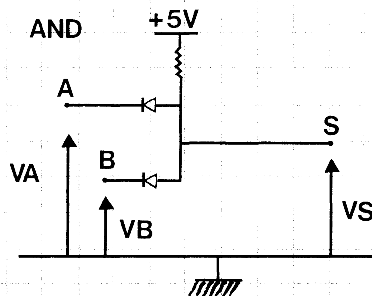
REPRESENTATION TRADITIONNELLE

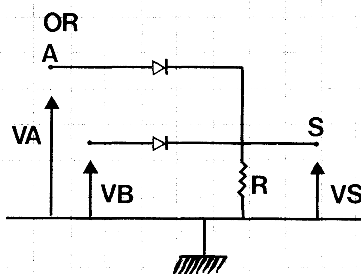
Chaque type de fonction a une représentation différente sur les plans, par exemple dans ceux du TRM :



SCHEMA ELECTRIQUE

Pour les amoureux de la complication, voici les schémas les plus simples :





CHANGEMENT DE BASES

Nous avons vu qu'il n'y a que deux représentations d'état sur un ordinateur : haut-bas ou 1-0 selon la convention.
Il faut donc trouver un moyen de représenter un nombre quelconque avec ces deux états. Notre système de numération prend en compte 10 états représentés par les nombres de 0 à 9. Puisque l'on veut représenter deux états, logiquement on utilisera la base 2, soient les nombres de 0 à 1. En base 2, le chiffre 2 n'existe pas et est remplacé par les nombres 10, de même qu'en base 10, le chiffre (9 + 1) n'existe pas et est remplacé par le nombre 10.
Soit un nombre est base 10 : A B C D où chaque lettre représente un chiffre. Ce nombre est égal à :
A^1000 + B^100 + C^10 + D^1 soit
A x 10^3 + B x 10^2 + C x 10^1 + D x 10^0

On généralise ceci en : soit un nombre A B C D en base N, sont équivalent en base 10 est :
AxN 3 + BxN 2 + CxN 1 + DxN 0
D'où, un nombre A B C D en binaire (N = 2) sera égal à :
Ax2 3 + Bx2 2 + Cx2 1 + Dx2 0 soit,
Ax8 + Bx4 + Cx2 + Dx1

Donc, si on considère un nombre en binaire constitué de 8 chiffres ou bits : ABCDEFGH, son équivalent en décimal sera : Ax128 + Bx64 + Cx32 + Dx16 + Ex8 + Fx4 + Gx2 + Hx1 chaque lettre valant 0 ou 1. Remarquez que si A = 0 et toutes les autres lettres valent 1, (01111111) le nombre sera 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127. Si seul A vaut 1, (10000000), alors ce nombre vaudra 128. Donc nous en déduisons que si un nombre en binaire a le bit 7 (= A) allumé, il vaudra moins 128. De même si on veut convertir un nombre N décimal en binaire, nous savons s'il est supérieur ou égal à 128 que le bit 7 est à un. Pour connaître la valeur des bits suivants, on regardera si la valeur correspondant aux 7 premiers bits (N-128x(N > 127)) est supérieur à 64 et on recommence jusqu'au bit 0, qui se trouve le plus à droite. Le mieux est de faire un peu d'entrainement, en s'aidant des deux routines publiées avec l'article « initiation au LH-5801 ».

FONCTIONS LOGIQUES EN LM

D'origine, le LH-5801 ne connait que :
OR mnémonics ORI ORA
AND mnémonics ANI AND
XOR mnémonic EOR

On peut facilement faire un NOT A en faisant EOR A,255 (FF en hexadécimal), ce qui a pour effet d'inverser tous les bits de A. Toutes ces fonctions travaillent donc sur 8 bits d'un seul coup. Grâce à elles, on peut reconstituer par combinaison les 3 absentes : NOR NAND et NXOR.

REPRESENTATION D'UN NOMBRE EN BASIC

Les fonctions BASIC OR, AND et NOT permettent de travailler sur des nombres 16 bits. En fait le BASIC utilise le 16 bits signés, c'est à dire que si le 16e bit est à 1, on dira que ce nombre est négatif.

HEX	0000	7FFF	8000	FFFF
DEC	0	32767	32768	65535
DEC signé	0	32767	-32768	-1

Voici les formules pour la conversion d'un système à l'autre : dans le nombre
CONVERSION 16 BITS EN 16 BITS SIGNES
A-(A>32767)x65536
CONVERTION 16 BITS SIGNES EN 16 BITS
A+(A<0)x65536

LES FONCTIONS LOGIQUES EN BASIC

Elles sont au nombre de trois :
OR AND et NOT
Attention ! Elles travaillent en 16 bits signés. Ainsi le NOT 0 donnera -1 ce qui correspond bien à 65535 en 16 bits. La syntaxe est celle ci :
A OR B
A AND B
NOT A

SIMULATION DU XOR

C'est en effet le seul qui manque pour pouvoir reconstituer le 7 fonctions décrites.
Nous allons voir les théorèmes fondamentaux qui régissent cet algèbre particulier appelé algèbre de BOOLE. Au lieu d'écrire à chaque fois le nom de la fonction en entier, on emploi des signes mathématiques qui se rapprochent de la fonction décrite :
OR +
AND .
XOR
Pour indiquer NOT, on met une barre sur la fonction à inverser :
NOT A = A
A NOR B = A + B

FORMULES

1) A + A = 1
A.A = 0
2) A + A = A.A = A
3) A + B = B + A
4) A + (B + C) = (A + B) + C
A.(B.C) = (A.B).C
5) A.(B + C) = A.B + B.C
6) A + (A.B) = A
7) A.(A + B) = A
8) A + B = A.B — 8 et 9 : Formules de
9) A.B = A + B — De Morgan

et enfin :
A + B = (A.B).(A + B)
= (A + B).(B + A)
= A.(A + B) + B.(A + B) (5)
= (A.A) + (A.B) + (B.A) + (B.B) (5)
= (A.B) + (B.A) (1)

donc :
A XOR B = (NOT A AND B) OR (NOT B AND A)
Pour faire un A NOR B, il suffit de faire NOT (A OR B)

UTILISATION EN INFORMATIQUE

Outre le fait absolument négligeable que cela permet de construire les ordinateurs, nous allons voir les utilisations courantes de ces fonctions, à l'exception des tests en BASIC qui seront traîtés à part.

xOR sert à allumer un bit. Par exemple :
ORA NN allume sur A les bits de NN qui sont à 1 et ne touche pas aux autres.
ORA FF met A à FF (255), tous les bits sont allumés.
ORI (NNNN), FF met l'octet n°NNNN à la valeur FF (255)
xAND sert à éteindre un bit.
AND NN éteint sur A les bits de NN qui sont eux mêmes éteints et ne touche pas aux ANI (NNNN), 00 met l'octet n°NNNN à 0. Très pratique.
xEOR sert à inverser l'état d'un bit.
EOR NN inverse sur A les bits de NN allumés et ne touche pas aux autres.

On s'en sert pour l'inversion vidéo ou faire des drapeaux (FLAG) sur un bit, qui prenne une valeur (0 ou 1) pour un nombre pair de passage et une autre (son complétement) pour un nombre impair.

(c) 1986 Marc GIRONDOT

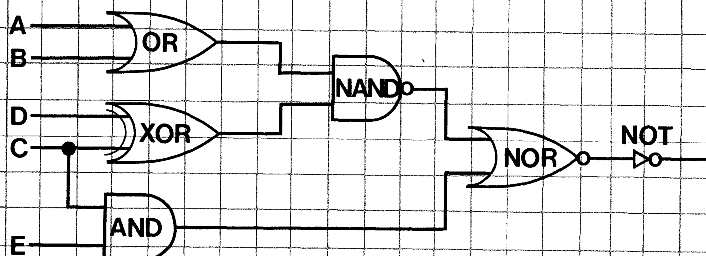
Renseignements tirés du cours de physique de SNV1

Etude d'une table de vérité

PC 1500 + 4Ko +CE 150 en option

Ce programme permet d'étudier la table de vérité d'une fonction booléenne. Il pourra être utilisé avec profit en électronique.

Par exemple, on peut étudier ce circuit.



Il comporte toutes les sortes de fonctions que le programme comprend à l'exception de NXOR qui est représenté par la fonction XOR suivi d'un petit rond. Il faut donc rentrer cette fonction dans le PC1500. Faites RUN, la séquence FTC : apparait, rentrer alors la fonction en suivant ces règles :

• chaque fonction élémentaire doit être entourée de parenthèse.

• Les courants en entrée sont désignés par une lettre A à H. Les courants 0 et 1 sont interprétés.

• La fonction NOT n'a qu'un paramètre placé derrière elle.

• Les autres fonctions ont deux paramètres, l'un devant la fonction, l'autre derrière.

• Un nom de fonction est toujours entouré d'un espace de chaque côté.

Il convient donc de décomposer le circuit.
On a (A OR B)
et (C XOR D) reliés par une porte NAND
donc ((A OR B) NAND (C XOR D)) qui est relié à ((ANDE) par une porte NOR donc (((AOR B) NAND (C XOR D)) NOR (CANDE)) puis un NOT sur tout cela donc :
(NOT (((A OR B) NAND (C XOR D)) NOR (CAND E)))

A	B	C	D	E	
0	0	0	0	0	1
				1	1
			1	0	1
				1	1
		1	0	0	1
				1	1
			1	0	1
				1	1
	1	0	0	0	1
				1	1
			1	0	0
				1	0
		1	0	0	0
				1	1
			1	0	1
				1	1
1	0	0	0	0	1
				1	1
			1	0	0
				1	0
		1	0	0	0
				1	1
			1	0	1
				1	1
	1	0	0	0	1
				1	1
			1	0	0
				1	0
		1	0	0	0
				1	1
			1	0	1
				1	1

Notez que l'ordre des arguments n'a pas d'influence sur le résultat (commutativité).
La visualisation peut se faire de plusieurs façons :
* à l'écran : si pas CE 150
ou si CE 150 et PRINT sur.
affichage des valeurs des courants à gauche et du courant en sortie à droite. Pour voir les noms des courants se superposer à leur valeur, appuyer sur ↕ et garder le appuyé sinon vous retournez aux valeurs. Appuyez sur une autre touche pour continuer l'étude.
* sur la CE150 : si CE150 et PRINT sur P
Noter qu'il y a deux types de tableau suivant que vous utilisez moins de 6 ou plus de 5 variables différentes. Cela est dû à une contrainte de l'imprimante.

(c) 1985 Marc GIRONDOT
et le club.

```
5:CLEAR :DIM A$(1)*80,A(10):INPUT "FCT:";A$(0):GOTO 7
6:GOTO 5
7:LP=1:IF PEEK# &B00E=255LET LP=0:GOTO 10
8:LP=PEEK# &B00EAND 32:IF LP=0GOTO 10
9:OPN :TEXT :CSIZE 1:COLOR 0:LPRINT A$(0):LPRINT :
   LPRINT
10:A$(0)="#"+A$(0)+"#":ON ERROR GOTO 999
15:Z=LEN A$(0):I=0:J=1
20:REM TESTE SI MEME NB DE ( QUE DE ) ET AU MOINS 1 ()
30:X=0:Y=0:FOR W=1TO Z
35:X=X+(MID$ (A$(0),W,1)=")"):Y=Y+(MID$ (A$(0),W,1)="("):
   NEXT W
40:IF X<>YOR X+Y=0PRINT "PARENTHESES !!!":GOTO 5
45:REM NOM DES VARIABLES UTILISEES DANS A(N)
50:Y=0:X=0:FOR W=1TO Z:IF MID$ (A$(0),W,1)<"A"GOTO 60
54:IF MID$ (A$(0),W+1,1)>="A"OR MID$ (A$(0),W-1,1)>="A"
   GOTO 60
55:A(X)=ASC MID$ (A$(0),W,1)-64:X=X+1
60:NEXT W
61:REM TESTE SI BONNES VARIABLES
62:FOR W=0TO X-1:IF A(W)<=8NEXT W:GOTO 70
65:PRINT "ERREUR DE NOMS":GOTO 5
70:REM TESTE SI UNE VARIABLE EST UTILISEE PLUSIEURS FOIS
80:W=0
85:IF X=1GOTO 130
90:FOR V=W+1TO X:IF A(V)<>A(W)NEXT V:GOTO 120
100:Y=V+1:V=X:NEXT V:FOR V=YTO X:A(V-1)=A(V):NEXT V:X=X-1:
   GOTO 85
120:W=W+1:IF W<X-1GOTO 85
130:REM REMPLACE LES 0 ET 1
140:FOR W=1TO Z:IF MID$ (A$(0),W,1)="1"LET A$="J":GOSUB "R
   EM"
145:IF MID$ (A$(0),W,1)="0"LET A$="I":GOSUB "REM"
147:NEXT W
150:REM TABLEAU DES ENTREES
151:IF LP=0GOTO 295
152:IF X>5TEXT :CSIZE 1:COLOR 1:FOR W=0TO X-1:LPRINT
   CHR$ (A(W)+64);" ";:NEXT W
153:IF X>5LPRINT :FOR W=0TO X-1:LPRINT "--";:NEXT W:
   LPRINT :GOTO 295
160:GRAPH :SORGN :CSIZE 1:U=INT ((2^X)+.1)
170:LINE (0,0)-(X*15,-10),,1,B
180:FOR W=15TO (X-1)*15STEP 15:LINE (W,0)-(W,-10):NEXT W
190:FOR W=0TO X-1:LINE -(W*15+5,-8),9:LPRINT CHR$ (A(W)+64
   ):NEXT W
200:LINE -(0,-15),9,0:SORGN
210:LINE (0,0)-(15*X,-U*10),0,,B
220:FOR W=15TO (X-1)*15STEP 15:LINE (W,0)-(W,-U*10):NEXT W
230:FOR W=1TO X:FOR Y=1TO INT (2^W-.9)
240:LINE (15*(W-1),(-10*Y*U)/INT (2^W+.1))-(15*W,(-10*Y*U)
   /INT (2^W+.1)):NEXT Y:NEXT W
250:FOR W=1TO X:A$="0":FOR Y=1TO INT (2^W+.1)
255:AA=(-10*Y*U)/INT (2^W+.1)
260:LINE -(15*(W-1)+5,(AA-(AA/Y)/2)-3),9:LPRINT A$:A$=
   CHR$ (ASC A$+1):IF A$="2"LET A$="0"
270:NEXT Y:NEXT W
280:LINE (X*15+5,0)-(X*15+20,-U*10),0,3,B:SORGN
290:FOR W=1TO U-1:LINE (0,-W*10)-(15,-W*10):NEXT W
295:AC=0
300:A$(1)=A$(0):Y=1
305:W=0:Z=LEN A$(1)
310:W=W+1:IF W=ZGOTO 600
320:IF MID$ (A$(1),W,1)="("LET AA=W
330:IF MID$ (A$(1),W,1)=")"LET AB=W:GOTO 350
340:GOTO 310
350:A$=MID$ (A$(1),AA+1,AB-AA-1)
360:IF LEFT$ (A$,1)=" "LET A$=RIGHT$ (A$,LEN A$-1):GOTO 36
   0
370:IF RIGHT$ (A$,1)=" "LET A$=LEFT$ (A$,LEN A$-1):GOTO 37
   0
380:IF LEFT$ (A$,3)="NOT"LET V1=ASC RIGHT$ (A$,1)-64:
   GOTO "NOT"
390:V1=ASC LEFT$ (A$,1)-64:V2=ASC RIGHT$ (A$,1)-64
395:A$=MID$ (A$,2,LEN A$-2)
400:IF LEFT$ (A$,1)=" "LET A$=RIGHT$ (A$,LEN A$-1):GOTO 40
   0
410:IF RIGHT$ (A$,1)=" "LET A$=LEFT$ (A$,LEN A$-1):GOTO 41
   0
420:GOTO A$
500:"STO"A$(1)=LEFT$ (A$(1),AA-1)+CHR$ (74+Y)+RIGHT$ (A$(1
   ),Z-AB)
510:@(Y+10)=VR:Y=Y+1:IF Y=12PRINT "PILE SATUREE":GOTO 5
520:GOTO 305
600:IF LP=0GOTO 650
601:IF X>5COLOR 0:FOR W=0TO X-1:LPRINT STR$ (@(A(W)));" ";
   :NEXT W:COLOR 2:LPRINT "  ";VR:GOTO 610
609:LINE -(5,-AC*10-8),9,2:LPRINT STR$ VR:AC=AC+1
610:U=X-1
620:@(A(U))=@(A(U))+1:IF @(A(U))<>2GOTO 300
630:@(A(U))=0:U=U-1:IF U<>-1GOTO 620
640:IF LPTEXT :LF 5
645:END
650:WAIT 0:FOR W=0TO X-1:PRINT STR$ (@(A(W)));:NEXT W:
   PRINT VR
660:A$=INKEY$ :IF A$=""GOTO 660
670:IF INKEY$ <>CHR$ 9WAIT :CLS :BEEP 1:GOTO 610
675:BEEP 1
680:FOR W=0TO X-1:PRINT CHR$ (A(W)+64);:NEXT W
690:IF INKEY$ =CHR$ 9GOTO 690
695:CLS :GOTO 650
700:"REM"A$(0)=LEFT$ (A$(0),W-1)+A$+RIGHT$ (A$(0),Z-W):
   RETURN
999:PRINT "SYNTAXE !!":GOTO 5
1000:"NOT"VR=1-@(V1):GOTO "STO"
1010:"OR"VR=@(V1)OR @(V2):GOTO "STO"
1020:"AND"VR=@(V1)AND @(V2):GOTO "STO"
1030:"XOR"VR=(@(V1)+@(V2))AND 1:GOTO "STO"
1040:"NOR"VR=1-(@(V1)OR @(V2)):GOTO "STO"
1050:"NXOR"VR=1-((@(V1)+@(V2))AND 1):GOTO "STO"
1060:"NAND"VR=1-(@(V1)AND @(V2)):GOTO "STO"
```

BASIC ET RECURSIVITE

**Voici un article qui vous fait part d'un style de programmation que j'affectionne beaucoup... la programmation récursive.
Même si ce style de programmation est destiné à des machines permettant la définition de variables locales, de procédures et de fonctions évoluées, le but de mon article est de prouver qu'elle est aussi très facilement concevable sur des machines qui ont un basic « standard », moyennant quelques « règles » simples.
Les exemples fournis sont réalisés sur les PC-1500 mais il n'y a normalement aucun problème en ce qui concernerait une adaptation sur d'autres machines (1)**

LA PROGRAMMATION RECURSIVE

J'ai longtemps cru qu'un ordinateur ne faisait que ce qu'on lui demandait de faire... ce n'est pas tout à fait vrai.
J'étais persuadé que les programmes n'étaient qu'une suite d'instructions à exécuter, avec de temps en temps des boucles et des tests qui tenaient compte des résultats obtenus...
... mais il est une autre programmation, moins courante et plus passionnante encore : la programmation récursive.
La programmation traditionelle est basée sur la décomposition d'un problème complexe en une suite d'opérations élémentaires qui seront exécutées séquentiellement.
La programmation récursive elle, suppose que le problème complexe peut se ramener à un problème élémentaire appelé cas de base. Elle nous demande aussi de trouver une relation simple qui y conduit. Elle ne se soucie pas de la notion du temps.
La relation simple qui conduit au cas de base est en fait un lien entre le problème complexe et le même problème d'un degré de complexité plus petit. Pour y voir plus clair, illustrons la différence entre ces deux concepts par un exemple.
Ecrire un sous-programme qui étant donné un nombre N, positif ou nul, imprime à l'écran, la suite de nombres N (N – 1).. 2 1 1 2.. (N – 1) N.

Par exemple,
pour N = 4, on obtient 4 3 2 1 1 2 3 4.
pour N = 0, on obtient rien du tout.
Le premier programme qui vient à l'esprit se déroule dans le temps et imprime successivement les valeurs de N à 1 puis de 1 à N... c'est la programmation traditionnelle.

Le programme aura l'allure suivante,

```
10 INPUT « SUITE = » ; S
 20  WAIT 0
     : N = S
     : GOSUB 100
 30 : WAIT : PRINT : END
100  REM
110  IF N = 0 THEN GOTO 130
120  FOR I = N TO 1 STEP – 1
     : PRINT I ;
     : NEXT I
     : FOR I = 1 TO N
     : PRINT I ;
     : NEXT I
130  RETURN
```

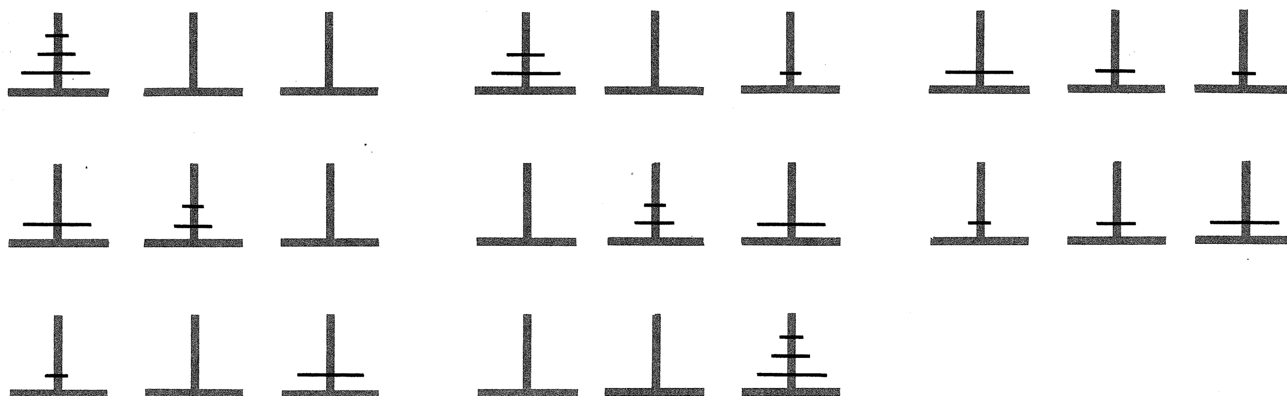
La programmation récursive recherche un cas de base : dans notre exemple, il me semble évident que c'est le cas de la suite « vide », celle qui correspond à N = 0 où il ne faut rien faire !
Il ne nous manque qu'à exprimer notre problème complexe - écrire la suite N - en fonction du même problème de complexité plus petite - écrire la suite (N – 1).
Pour trouver cette relation, il suffit de se dire qu'on sait résoudre le problème de complexité plus petite... elle vient alors tout de suite :
écrire la suite N, c'est (si N est plus grand que 0),
(a) écrire N,
(b) écrire la suite (N – 1),
(c) écrire N.
On remarque, comme prévu, que le sous-programme aura besoin de lui-même pour écrire la suite N – 1 !!!
En programmation récursive si on utilise une machine qui n'a pas un basic procédural, il est un certain nombre de règles qu'il faut accepter dans un premier temps, et qu'il est intéressant de comprendre par la suite.
Il faut avant tout,
(0) remplacer toutes les variables du sous-programme par des variables indicées, càd des tableaux. La dimension de ceux-ci étant égale la complexité du problème initial.
Il faut ensuite,
(1) avant l'appel du programme principal au sous-programme, placer R = 0.
(2) au début du sous-programme, placer R = R + 1.
(3) à la fin du sous-programme, placer R = R – 1.
(4) indicer toutes les variables sur lesquelles le programme travaille effectivement par R.
(5) indicer tous les paramètres que l'on passe au sous-programme par R + 1.
(6) indicer tous les paramètres qui reviennent du sous-programme par R – 1.

Voici le programme	règles,	commentaires,
10 INPUT "SUITE = "; S		
: DIM N(S + 1)		
20 WAIT 0		
: R = 0	(1)	
: N(R + 1) = S:GOSUB 100	(5)	
30 WAIT :PRINT : END		
100 R = R + 1	(2)	On écrit le cas de base...
110 IF N(R) = 0 THEN GOTO 130	(4)	...et notre relation
120 PRINT N(R);	(4)	(a)
: N(R + 1) = N(R) – 1:GOSUB 100	(4) et (5)	(b)
: PRINT N(R);	(4)	(c)
130 R = R – 1	(3)	le tout en BASIC !
: RETURN		

Vous pouvez vérifier que ce programme fournit le résultat attendu... et le comparer avec celui que nous avions réalisé en programmation traditionnelle.
Il faut bien comprendre la ligne 120 - application des règles (4) et (5) - et qui donc
(a) imprime le N courant,
(b) passe comme paramètre N, la valeur de N courant moins un,
(c) puis imprime encore le N courant.
L'intérêt de la programmation récursive n'apparaît que dans des exemples plus subtils...
Vous avez sûrement déjà écrit un programme qui cherchait une solution à un problème que vous même ne saviez pas résoudre mais en tout cas, vous saviez comment faire résoudre à votre place le problème par votre machine. Par exemple, pour une recherche de racines de fonction, la méthode de la tangente !
Avec la programmation récursive, vous pouvez résoudre des problèmes sans avoir aucune idée de la manière avec laquelle l'ordinateur devra trouver la solution... on comprend pourquoi la récursivité est à la base des langages d'intelligence artificielle !
Pour vous convaincre, voici l'exemple classique des tours de HANOI.
On imagine 3 piquets, sur le premier (la tour de départ) se trouvent N disques de dimensions différentes. Ils sont empilés par ordre de grandeur, le plus grand se trouvant en dessous. Le problème consiste à placer ces disques sur la troisième tour (la tour d'arrivée) dans le même ordre, en les prenant un à un, et en passant par des tours intermédiaires, sans jamais placer un disque plus grand sur un plus petit.

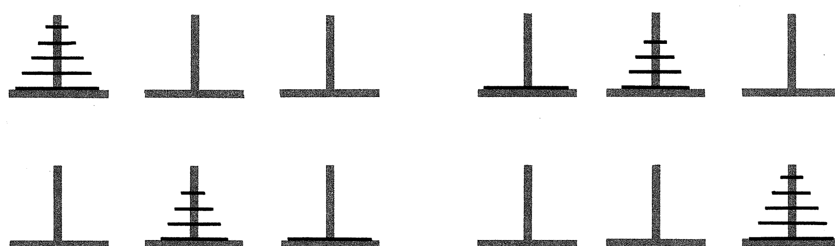
Par exemple, pour N = 3, on a la succession suivante : de 1 à 3, de 1 à 2, de 3 à 2, de 1 à 3, de 2 à 1, de 2 à 3 et de 1 à 3.

Graphiquement,



N'ayez pas peur ! C'est très simple à programmer, car comme on l'a dit, il n'est pas nécessaire de savoir comment s'y prendre !!! Il suffit de chercher un cas de base et une relation qui y conduit.
Si la tour de départ ne contenait qu'un seul disque, nous pourrions résoudre le problème : il suffirait de prendre un disque de la tour de départ, et de le placer sur la tour d'arrivée... voilà notre cas de base.
Il ne nous reste qu'à exprimer notre problème complexe - déplacer N disques - en fonction de ce même problème de complexité plus petite - déplacer (N – 1) disques.
Pour trouver cette relation, il suffit de se dire qu'on sait résoudre le problème de complexité plus petite... elle vient alors tout de suite.
Déplacer N disques de la tour de départ à la tour d'arrivée en passant par une tour intermédiaire, c'est (si N est plus grand que 1),
(a) en déplacer N – 1 de la tour de départ à la tour intermédiaire en passant par la tour d'arrivée,
(b) en déplacer un de la tour de départ à la tour d'arriver,
(c) redéplacer les N – 1 de la tour intermédiaire à la tour d'arriver en passant par la tour de départ !

Graphiquement, pour N = 5,



On peut alors écrire le programme presque directement.
On appelera D le numéro de la tour de départ, A celui de la tour d'arriveé, V celui de la tour intermédiaire et Q la quantité de disques à déplacer. Ces variables seront indicées comme le demandent les règles (4) et (5).

Voici le programme, commentaires,

```
10  INPUT « QTE DISQUES (S)=":N
    : DIM D(N), A(N), V(N), Q(N)            règle (0)
20  R=0
    : D(R+1)=1:V(R+1)=2:A(R+1)=3            on part de 1 vers 3 via 2
    : Q(R+1)=N:GOSUB 100                    avec N disque, règle (5)
30  END
100 R=R+1                                   règle (2)
110 IF Q(R)=1PRINT "DE"; D(R);
    "-A";                                   On écrit le case de base...
    A(R)
    : GOTO 150                              et notre relation
```

programme, commentaires,

```
120 D(R+1)=D(R): V(R+1)=A(R):
    A(R+1)=V(R)                             (a)
    : Q(R+1)=Q(R)-1: GOSUB 100              règle (4) à (5)
130 D(R+1)=D(R): A(R+1)=A(R):               (b)
    : Q(R+1)=1: GOSUB 100
140 D(R+1)=V(R): V(R+1)=D(R):
    A(R+1)=A(R)                             (c)
    : Q(R+1)=Q(R)-1 : GOSUB 100             le tout en BASIC !!!
150 R=R-1                                   règle (3)
    : RETURN
```

Voilà... Puissant non ?
On ne pourrait terminer sans donner une application mathématique - les sharpentiers aiment les maths !
Nous allons généraliser un programme qui calcule les intégrales simples du type

$$I = \int_a^b y(x)\,dx$$

par la méthode de GAUSS, à un programme qui calcule les intégrales n-uples du type

$$I_n = \int_a^b \int_{a(x_1)}^{b(x_1)} .. \int_{a(x_1,...,x_{n-1})}^{b(x_1,...,x_{n-1})} y(x_1,X_2,...X_n)\,dx_n \ldots dx_2\,dx_1$$

Voici le programme archi-classique d'intégration numérique en 3 points d'une fonction d'une variable y=y(x). On trouve une description de l'algorithme dans tout syllabus d'analyse numérique.
Les paramètres nécessaires à l'intégration sont A, B (les bornes d'intégration) et Q le nombre entier de sous intervalles (augmenter Q allonge le calcul et améliore le résultat).

Programme, commentaires

```
10  GOSUB 12:PRINT "ITG",I:END      La sous routine en ligne 12
12  GOSUB 20                        va elle même lire les
                                    paramètres.
    : U=(B-A)/2/Q:Z=A+U             Initialisation du paramètre
    : I=0                           qui sera retourné.
14  FOR K=1 TO Q
    : X=Z-U*SQR .6 : GOSUB
    30 :W=Y
    : X=Z+U*SQR . 6 : GOSUB
    30 :W=W+Y
16  X=Z : GOSUB 30 :
    I=I+5*W+8*Y                     Calcul de gauss en 3 points.
    : Z=Z+2*U
    : NEXT K
18 I=I/9:RETURN
20  A=0, B=1, Q=1 : RETURN          Ici, on place les paramètres.
30  Y=X*X                           Ici, la fonction.
32  RETURN
```

Cet exemple calcule donc $I = \int_0^1 x^2\,dx$.

La généralisation aux intégrales n-uples demande un cas de base, c'est celui de l'intégrale simple d'une fonction Y donnée...
La relation qui lie une intégrale N-uple à une intégrale (N-1)-uple est la suivante (si N est plus grand de 1),
(a) une intégrale N-uple est l'intégrale d'une intégrale (N-1)-uple !!!
Vous voyez, avec l'habitude, les relations semblent de plus en plus simples...
Pour otpimiser nos programmes récursifs, nous allons modifier un peu la règle(0) et ajouter une dernière règle.
(0') il ne faut remplacer dans le sous-programmes que les variables du sous-programme dont la valeur doit être conservée après un rappel récursif de ce dernier.
(7) Quand on doit passer un paramètre dont la valeur est justement celle de notre variable R (celle qui sert à indicer les variables...) ce paramètre peut être supprimé (càd remplacé implicitement par R). On économise ainsi un tableau !... R joue alors un rôle double.
Dans notre programme, la règle (0') s'applique à A, B, Q et Y.
La règle (7) est d'application : en effet, il nous faut un paramètre qui donne l'indice ($_1$, $_2$,.... ou $_n$) de la variable (x_1, x_2... ou x_n) sur laquelle se fait l'intégration courante. Or, il se fait que ce paramètre aura précisement toujours même valeur que notre véritable R.

Programme, Commentaires

```
10  INPUT "N=";N
    : DIM U(N), Z(N), I(N), K(N),
    X(N), W(N)
    : R=0
    : GOSUB 12:PRINT "ITG", I(R):
    END                             La sous routine en ligne 12
12  R=R+1                           va elle même lire les paramè-
                                    tres qui changent ici pour
                                    chaque variable
    : GOSUB 19+R
    : U(R)=(B-A)/2/Q:
    Z(R)=A+U(R)                     d'intégration. (pour R, cfr
                                    règle (7))
    : I(R-1)=0                      Initialisation du paramètre
```

```
14 FOR K(R)=1 TO &                    qui sera retourné
   : X(R)=Z(R)-U(R)*SQR .6:
   GOSUB 30: W(R)=Y
   : X(R)=Z(R)+U(R)*SQR . 6:
   GOSUB 30 : W(R)=W(R)+Y
16 X(R)=Z(R): GOSUB 30:I(R-1)
   =I(R-1)+5*W(R)+8*Y
   : Z(R)=Z(R)+2*U(R)
   : NEXT K(R)                        même calcul que tantôt
18 I(R)=I(R)/9 : RETURN
20 A=0, B=4, Q=1: RETURN              Maintenant, on place les
21 A=0, B=SQR X(1), Q=1:
   RETURN                             A,B et Q des Xi en 20+i
30 IF R=N LET
   Y=X(2)  3+2*X(1)*X(2):
   RETURN                             Cas de base et
32 GOSUB 12 : Y=I(R): RETURN          relation entre les intégrales !!!
                                      (sous entendu, on passe R :
                                      règle (7))
```

Remarque

Les valeurs placées dans ce programme correspondent au calcul de

$$I_2=\int_0^4 \int_0^{sqrt(x_1)} x_2^3 + 2\, x_1\, x_2\, dx_2\, dx_1$$

Le résultat fourni ici est tout à fait correct, vu que nous sommes dans le cas où l'algorithme de Gauss s'applique particulièrement bien.(2)
Il reste encore bien des applications intéressantes, plus compliquées ou plus simples... les plus courageux d'entre vous peuvent écrire un programme récursif qui évalue le déterminant d'une matrice N x N.
Le cas de base est la matrice 1 x 1.
La relation des coffacteurs nous donne la relation de récursivité.
A bientôt, et bon amusement.

1 Cependant, certaines machines refusent un GOTO calculé, comme c'est le cas dans le dernier exemple, on remplacera alors le GOTO 19+R de la ligne 12 par ON R GOTO 20, 21, 22, 23, 24.

Enfin, d'autres basic ne supportent pas les boucles FOR-NEXT avec des tableaux... il suffit d'en revenir alors à la boucle classique avec un GOTO car

```
FOR I=1 TO 100 :..: NEXT I
```

est équivalent à

```
I=1
L1 : .... : I=I+1
IF <=100 THEN GOTO L1
```

En effet, l'algorithme marche très bien si les dérivées de la fonction que l'on intègre sont finies aux bornes d'intégration.

Frédéric BLONDIAU

LES SOUS-PROGRAMMES

**Dans presque tous les livres d'initiation au BASIC, il est dit que pour faire un bon programme, il faut faire de chaque tâche élementaire un sous-programme.
C'est donc une notion fondamentale qu'il convient de bien avoir saisie avant de se lancer dans de grandes applications. En fait, tous les langages informatiques utilisent des sous-programmes, le langage machine en est même particulièrement friand et les langages utilisant la notion de Procédure fonctionnent grâce à eux : C, PASCAL, FORTH...**

QU'EST CE QU'UN SOUS-PROGRAMME ?

Vous en utilisez tous, sans même vous en rendre compte, puisque l'on peut dire en première approximation que toute fonction BASIC est un sous-programme.
Ecrivez cette ligne :
10 : BEEP 1 : PRINT « FIN »

Comme vous le savez (sinon arrêtez de lire, cet article est de trop haut niveau pour vous) cette ligne exécute un BEEP puis écrit FIN. Le programme en ROM (mémoire morte) qui se charge d'interpréter les mots BASIC en bianire (d'où son nom d'interpréteur) pour qu'ils soient traités par le microprocesseur considère chaque mot BASIC comme un sous-programme :

```
10 : BEEP        1 : PRINT        « FIN »
       EXCUTION           AFFICHAGE
```

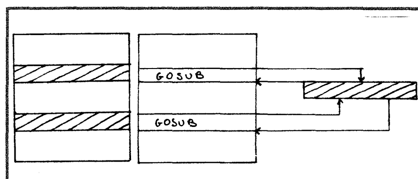
On voit la particularité d'un sous-programme, qui est de revenir tout de suite apès l'endroit d'où il a été appelé. Près à exécuter l'instruction (ou sous-programme) suivante. C'est ce que l'on pourrait appeler des sous-programmes implicites, en effet vous ne dites pas spécifiquement que vous utilisez un sous-programme mais c'est traduit comme tel.

LES SOUS-PROGRAMMES EXPLICITES

Quand vous employez l'instruction GOSUB, vous définissez une certaine partie de votre programme comme étant un sous-programme. Celui-ci se situera de la ligne dont le numéro est l'argument du GOSUB jusqu'au Premier RETURN trouvé, qui marquera la fin du sous-programme, et le retour a l'appel.
Pour s'y retrouver l'interpréteur BASIC utilise une zone de la mémoire pour stocker le numéro de la ligne et l'endroit où il se trouve dans cette ligne lors de l'appel du sous-programme. Pour savoir manier cette pile je vous conseille de reprendre votre SHARPENTIER n° 9 page 14 et de l'examiner.

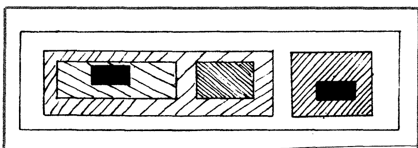
POURQUOI UTILISER UN SOUS-PROGRAMME

Dans les livres d'initiation, on conseille d'utiliser les sous-programmes pour augmenter la lisibilité. C'est vrai qu'un programme écrit en utilisant de nombreux sous-programmes est facilement compréhensible pour celui qui ne l'a pas fait.
On utilise aussi souvent un sous-programme lorsqu'une séquence se répète plusieurs fois à l'intérieur d'un programme. On isole alors cette sequence en la faisant devenir un sous-programme, et à chaque fois que cette séquence était écrite, on la remplace simplement par un GOSUB ce qui se traduit par un gain considérable d'octet.



LES SOUS-PROGRAMMES IMBRIQUES

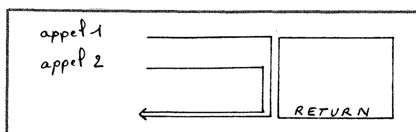
On a vu qu'à partir d'un programme principal on pouvait appeler un sous-programme. Mais on peut aussi appeler un sous-programme à partir d'un autre sous-programme. Dans les livres d'initiation on représente les sous-programmes par des rectangles qui sont imbriqués mais qui ne doivent jamais se croiser. Ceci n'est en fait qu'une approximation car avec cette représentation des sous-programmes on ne peut exprimer le fait que le sous-programme représenté par le rectange plein peut-être appelé par les deux sous-programmes dans lesquels il se trouve et cela sans qu'il soit besoin de le réecrire deux fois.



ASTUCES SUR LES SOUS-PROGRAMMES

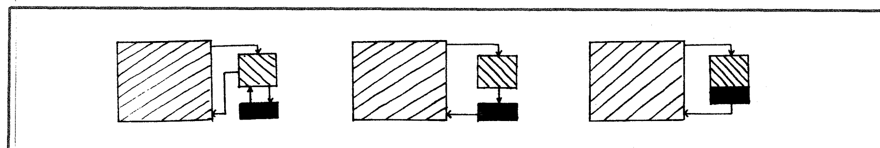
LE COMMENCER EN COURS

On peut considérer un sous-programme comme étant un tout, mais en fait il est souvent utile de le commencer au milieu pour en sauter une partie qui ne servirai qu'en certaines occacions.



GOTO AU LIEU DE GOSUB

Dans certains cas il est préférable de remplacer un GOSUB par un GOTO, ce qui fait gagner quelques octets. Si vous avez une séquence de type :
GOSUB « LIGNE 1 » : RETURN
vous pouver sans problème la remplacer par une séquence du type :
GOTO « LIGNE 1 »
et lorsque l'interpréteur rencontrera le RETURN signalant la fin du sous-programme dénommé « LIGNE 1 », il ne reviendra pas après le GOTO « LIGNE 1 », mais aprés cette séquence.
On peut même être plus astucieux encore en placant le sous-programme « LIGNE 1 » juste après la fin du sous-programme où se trouve l'appel GOTO « LIGNE 1 » et éliminer celui-ci. On se trouve alors dans le cas « LE COMMENCER EN COURS »



SOUS-PROGRAMMES AU DEBUT

Cette bonne habitude nous vient du temps où l'on programmait sur PC-1211. Celui-ci était, en effet, connu pour sa lenteur et l'on recherchait par tous les moyens des astuces pour rendre les programmes un peu moins lents. On avait trouvé que l'accès à un sous-programme est plus rapide s'il se trouve en tête du programme, et celà s'est avéré toujours vrai avec le PC-1500.
En fait ceci s'explique tout à fait, puisque quand le PC-1500 recherche où se trouve une ligne, il le fait en commencant par le début du programme.

LIEN AVEC LE LM : PSH POP

Vous n'espériez tout de même pas lire un article entier sur le PC-1500 sans trouver un mot sur le langage machine !
Tout ce qui a été dit précédemment est applicable sans réserve au langage machine en remplacant les :
GOSUB en SJP (Subroutine JumP)
RETURE en RTN
GOTO en JMP (JumP)

Mais on peut faire nettement mieux en langage machine, car l'on peut jouer sur la pile où sont stockées les adresses de retour.

Lorsque vous faites un SJP, le microprocesseur stocke dans un endroit de la mémoire appelé la pile l'endroit de la mémoire où se trouve l'instruction suivante, et s'en va à l'adresse correspondant aux paramètres du SJP. Lorsqu'il trouve un RTN, il reprend dans la pile le nombre qui s'y trouve et va à l'adresse y correspondant. On dit qu'une telle pile est une pile LIFO : Last In First Out ou pour les Anglophobes : dernier rentré premier sorti, ceci s'appliquant bien sûr aux nombres qui y sont stockés. Mais lorsque vous êtes dans le sous-programme, vous pouvez modifier la pile. Avec PSH vous rentrerez des données, comme si vous les posiez sur celle préexistantes sans rien altérer. Avec POP vous reprendrez ce qu'il y a en haut de la Pile. Donc si vous mettez quelque chose sur la pile, il ne faudra pas oublier de l'enlever avant d'exécuter le RTN, sinon celui-ci prendrai cette donnée pour l'adresse d'où il a été appelé, ce qui conduit à des résultats inattendus. Mais celà peut-être voulu, il faut faire très attention à ce que l'on fait.

Par exemple nous allons stimuler un JMP à une adresse se trouvant en U :
PSH U U se trouve en haut de la pile
RTN prend la valeur du haut de la pile et se rend à cette adresse , on peut de même annuler l'effet d'un SPJ par un POP.
POP X et vous aurez dans X l'adresse où le programme serait allé s'il avait rencontré un RTN.

Vous pouvez faire :
POP V Pour ne modifier aucun registre. Le SPJ par lequel votre programme est venu à cette adresse aura eu le même effet qu'un JMP.
Vous pouvez combiner ces POP et ces PSH en faisant :
POP V
PSH U
RTN

Vous remplacez alors l'adresse de retour par celle se trouvant en U.

(c) 1985 Marc GIRONDOT

ASTUCES

ASTUCE 1

BEEP..ZARRE

Encore une particularité de la ROM du PC-1500 qui n'a pas fini de nous étonner.

C'est à la mise au point du super-CALL que cette particularité a été découverte. Ce n'est pas grand chose, cela n'a aucun intérêt, mais puisque la chose est connue, autant en faire profiter tout le monde !
Alors qu'un BEEP a toujours besoin d'un paramètre, on peut ne pas lui en mettre si on se trouve en BEEP OFF. Inutile n'est-ce pas !
10 : BEEP ON : BEEP :A=1
provoque une ERROR 1, mais
10 : BEEP OFF : BEEP : A=1
marche tout à fait bien. La bogue en ROM est à l'adresse C684.
Pour la corriger, il aurait fallu faire :
E5D7 : BII (786B), 01 si BEEP OFF, Z est baissé
E5DB : BZR +73 (E650) si BEEP OFF, va en E650
....
E650 : JMP C684 cherche le : suivant
....
C684 : DEC Y cette instruction est de trop car dans ce cas là il revient sur le deuxième code du mot-clé alors qu'il était bien positionné.
Mais si les concepteurs avaient voulu corriger cette omission, ils auraient utilisé de la mémoire, et finalement, ce n'est pas si grave.

(c) 1986 Marc GIRONDOT

ASTUCE 2

PROGRAMME « DELETE ZONE »

N'étant pas initié du LM, j'ai utilisé le programme « NEW » publié dans le SHARPENTIER n° 15 page 24 implanté à l'adresse &38C5 pour une utilisation différente de celle d'origine.

Ayant l'habitude d'utiliser un certain nombre de programmes mergés, le programme ci-joint me permet d'effacer très rapidement une zone mergées et de conserver les autres zones. La ligne 6 prend en compte le dernier programme de la zone BASIC et l'élimine simplement en changeant les adresses des pointeurs. Faire RUN et répondre par la première étiquette de la zone à effacer. Bien sûr il faut mettre en &38C5 le programme « NEW ». Si vous voulez le mettre autre part, il faut modifier le CALL &38C5 en ligne 7.

(c) 1986 Le club et Mr JACOB Patrice

PROGRAMME « DELETE-ZONE(M) ».

```
1 « W » CLEAR : A=PEEK &7865, B=PEEK &7866 : ON ERROR GOTO « W » : INPUT « Zone-DEL= » ; N$
2 IF N$=«» OR N$=« W » END
3 RESTORE N$ : I=256*(PEEK &78BE-128)+PEEK &78BF-3, J=I : RESTORE « W »
4 BEEP 1 : J=J+3=PEEK(J+2) : IF PEEK J<> 255 THEN 4
5 IF J> STATUTS 2-1 END
6 IF J= STATUS 2-1 LET I=I-1 : POKE &7867, INT(I/256), I-PEEK &7867*256, A, B : BEEP 3 : END
7 J=J+1 : POKE &7865, INT(J/256), J-PEEK &7865*256 : CALL &38C5, I : POKE &7865, A,B, : BEEP 3 :END
```

Remarque : CALL &38C5 lance le programme LM de pseudo NEW, I. I=Adresse LM+1 = nouvelle adresse de début de zone BASIC.

ASTUCE 3

NETTOYAGE EN RAM

Cette petite routine relogeable est utile lorsque l'on veut programmer agréablement à l'aide du moniteur SHARP.

```
VEJ CC, 65     CC 65  X=(7865-7866)
LDA V1         34     A=0
SIN X          41     (X)=0 X=X+1 <-+
CPI Xh, 70     4C 70  Xh=70 ?       |
BZR -05        99 05  non ----------+
RTN            9A
```

Pour l'utiliser, CALL (1[er] adresse).
La partie de mémoire comprise entre le début du BASIC et la fin de la mémoire sera mise à 0.

M. BOMBA Xavier

ASTUCE 4

LA PUISSANCE DES PC

Pour signifier une puissance de dix dans un calcul, on utilise la lettre E plutôt que le 10^habituel. Mais attention, le E peut donner des résultats étranges si on n'y prend garde, pour vous en convaincre, essayez :
1E4.3 ENTER
Et il s'affiche 1E 43 alors que l'on devrait trouver
10 4.3=19952.62315

De même, un 1E1234 ne devrait pas exister puisque les PC n'acceptent des puissances que jusqu'à 99, or il s'affiche : 1E 34.
Nous pouvons en déduire que si la puissance de 10 a plus de deux chiffres, le PC ne prend en compte que les deux derniers, en éliminant la virgule s'il s'en présente une.

Je n'ai pas encore pu déterminer où se situe la partie de la ROM incriminée, si vous avez des information, n'hésitez pas à nous les communiquer.

(c) 1896 Marc GIRONDOT

ASTUCE 5

NOTE SUR PC BLUES

Oui, le programme PC-BLUES paru dans le SHARPENTIER n° 15 fonctionne bien !
En fait, c'était plutôt le programme PLAY qui était incrimé. Si vous n'avez pas réussi à le faire tourner, c'est peut-être simplement parce que vous avez une version de ROM ancienne. Il suffit alors de remplacer la ligne 190 du listing 4 par :
190 IF A<>970 LET I=I+1 : GOTO 180

(c) 1986 Martin RAYROLLE

ASTUCE 6

PRECISIONS SUR SOFT TEXT

Voici les dernières astuces trouvées pour utiliser ce logiciel :
F1 : PEEK &78E4x256+PEEK &78E5−PEEK &78E0x256-PEEK &78E1
F2 : PEEK &78E2x256+PEEK &78E3−PEEK &78E4x256-PEEK &78E5

Une pression sur F1 vous donnera la mémoire utilisée et sur F2 la mémoire restante.

Pour ne sauver sur cassette que les codes définis selon l'imprimante :
CSAVE M PEEK &7863x256+1096, PEEK &7863x256+1235
Et pour les recharger :
CLOAD M

Pour mettre ce programme sur CE-161 et l'y laisser continuellement :
NEW 0
CLOAD M (comme normalement)
NEW &3801
CLOAD (seul le BASIC se chargera)
Avant utilisation il faudra faire :
POKE &7864, &40
que l'on pourra mettre en ligne 5.
Avant de retirer le module, faire comme indiqué page 19 du n° 17.

SUPER CALL

Le seul défaut du CALL est de ne pouvoir passer que très peu de paramètres. Nous allons faire mieux !

Les concepteurs du PC-1500 ont eu la bonne idée de permettre le passage de paramètres à un CALL. Voici les résultats :

VARIABLE NUMERIQUE

CALL (&nnnn), v
avec dans v la valeur &vvvv en 16 bits signés, (−32768, +32767)
X=&vvvv Y=&C897 U=&nnnn A=&00

VARIABLE ALPHANUMERIQUE

CALL (&nnnn), v$
Soit &aaaa l'adresse du premier octet de la variable en MEV
X=&aaaa Y=&C897 U=&nnnn A=Nb max de caractères dans v$

Remarquez que si vous utilisez des variables dimensionnées, il faut le faire avant le CALL, ainsi :
CLEAR
CALL &40C5,AA
provoque une erreur 7 car la variable AA n'a pas été définie. On fera donc :
AA=0
CALL &40C5,AA

Si on fait seulement un :
CALL (&nnnn)
X=&C86C Y=&C897 U=&nnnn A=&80
De plus, dans tous les cas, on aura dans la pile :
&C897
Adresse du premier octet après le CALL(: ou 80D)
&00
&C455
Les 00 et C455 sont dûs au traîtement interne du CALL.
Au retour du CALL, si il y a eu passage de paramètre, on peut demander au programme en LM de repasser des paramètres au BASIC, on s'arrangera pour cela à ce le CARRY soit levé (SEC). Si numérique : Charge la valeur U dans la variable définie, si alphanumérique : charge la chaîne de longueur A à l'adresse X dans la variable définie.

MIEUX ENCORE !

Le principal défaut de cette méthode est de ne pouvoir passer qu'une variable, et aucune expression algébrique. Nous allons y remédier.
Voyons tout d'abord dans quel état se trouve la machine à l'arrivée sur une routine de mot clé :
X=adresse de la routine
Y=adresse du premier octet suivant le code de la fonction
U=adresse dans la table suivant l'adresse de la routine
A=? (semble correspondre au type d'utilisation : RUN PRO)
Dans la pile :
&00
&C455

Maintenant, nous allons berner le système, on va lui faire croire qu'un CALL (&nnnn) est en fait un seul mot clé :
On va simuler un BEEP avec un CALL :
CALL &nnnn : 10 BEEP 10
Pointeur :
Position : pile Y
Il y a peu de différence, et nous allons les éliminer :
Après le CALL &nnnn, il faut faire :

```
POP  Y dans Y &C897
POP  Y dans Y l'adresse du :
INC  Y se positionne sur le 1
JMP &E5C1 se branche sur la routine BEEP.
```

Faites par exemple :

```
NEW &40CE
POKE &40C5, &FD, &1A, &FD, &1A, &54, &BA, &E5, &C1
10 : CALL &40C5 : ON : CALL &40C5 : 10, 20, 310
```

ce qui correspond à :

```
10 : BEEP ON : BEEP 10,20,310
```

On peut même être encore plus fort en utilisant le passage de paramètre pour dire quelle est la routine à utiliser :

```
POP Y   FD 1A   Y=&C897 inintéressant
POP Y   FD 1A   pointe sur le :
CPA VL  36   A=0 si variable numérique
BZS +03  8B 03 si A=0, bon
LDI UH,07  68 07  prépare l'erreur 7
VEJ EO  EO  erreur 7
INC Y  54  pointe sur l'octet suivant le :
STX P  FD 5E  se branche à l'adresse de la var.
```

On fera alors :

```
NEW &40D3
POKE &40C5, &FD, &1A, &FD, &1A, &36, &8B, 3, &68, 7, &E0, &54, &FD, &5E
10 : A=&E5C1−&FFFF−1 : CALL &40C5, A : ON : CALL &40C5, A : 10,20,310
```

Remarquez la convention de &E5C1 en 16 bits pour à l'arrivée dans la routine que l'on obtienne bien le &E5C1.
Il ne vous reste plus qu'à exploiter ces nouvelles possibilités et à nous envoyer vos découvertes.

CLASSEMENT

Cette routine en LM est destinée à être utilisée immédiatement dans tous les programmes de gestion de fichiers en BASIC.

En effet, elle simplifie toutes les procédures de tris qui sont particulièrement longues à l'exécution, et gourmandes en mémoire si on veut tenir compte de toutes les possibilités ou en faire une version un tant soit peu rapide. Notre version résout ces problèmes :

• Le LM permet une bonne rapidité, même si la méthode de tri utilisée est la plus rudimentaire (donc la plus facile à programmer en LM !)

• Elle envisage presque tous les cas. Voyons ceux-ci :

Un fichier est constitué de plusieurs rubriques qui sont stockées dans des variables dimensionnées de noms différents. Bien sûr, ceci est dans un cas général, on peut voir certaines adaptations selon les permormances recherchées. Donc, si l'on décide de trier ce fichier, il faut penser que le tri doit réamménager toutes les variables. Ainsi, si on gère un agenda avec nom et n° de téléphone dans deux varables différentes, il faut penser que si on change la place d'un nom, il faut faire suivre le n° de téléphone sous peine de déranger une personne non concernée à la première utilisation. Donc le programme le permet, puisque l'on peut définir 5 tableaux de variables en plus de celui sur lequel on fait le test. Ce dernier tableau peut être bien entendu n'importe lequel. De plus, il peut arriver dans certains cas que l'on ne désire modifier qu'une certaine partie du tableau. Il faudra donc définir une borne inférieure et une supérieure. On se servira aussi de cette fonction pour définir si les éléments sont stockés à partir de l'indice 0 ou 1. Et bien sûr, il faudra aussi définir si le tri doit être croissant ou décroissant.

Pour passer tous ces paramètres, on utilisera l'astuce décrite dans l'article SUPER-CALL. La syntaxe générale est : CALL & NNC5 : PI, DI, OR, TA1 (*), TA2 (*), TA3 (*), TA4 (*), TA5 (*), TA6 (*).

Les paramètres de TA2 (*) à TA6 (*) peuvent être tout ou partie ignoré. Le programme ne comprend que les tableaux ayant au maximum un nom de 2 caractères. De toute façon les autres signes sont ignorés par le PC-1500 en temps normal. De plus, il ne traîte pas les tableaux@(*) et@$ (*), ceux-ci ayant une position particulière en MEV système. Voici la signification des variables :

PI : premier indice

DI : dernier indice

OR : ordre, si supérieur ou égal à 0, tri croissant, sinon décroissant. Ou pour dire la même chose : si SGN OR = -1 tri décroissant et si SGN OR = 0 ou = λ1 tri croissant.

IMPLANTATION DE LA ROUTINE

Vous avez maintenant l'habitude du format d'impression à utiliser avec le moniteur SHARP. La colonne de gauche correspond à l'adresse, le code tout à fait à droite est un code de contrôle et les codes entre sont le programme. Il est listé pour être placé à l'adresse &C5. Sa fin sera alors en &29A. Il est fourni une routine de relogement qui ne reloge que le poids fort. L'adresse de début sera donc toujours : &NNC5. La fin sera logiquement en &N (N+2) 9A. Il faudra donc faire un NEW &N (N+2) 9B pour protéger ce programme d'une destruction accidentelle. La valeur &NN est donnée par l'octet &7863 en faisant PEEK &7863 on demande en effet le poids fort du premier octet de RAM, et le BASIC commence &C5 octets plus loin.

Faites tout d'abord le NEW adéquate selon votre version mémoire. Tapez le programme à l'adresse correspondant à l'adresse où il devra tourner, puis tapez le programme de relogement, RUN puis répondre à la question par l'adresse où se trouve le programme. Quelques secondes plus tard, 5 bips vous signalent que l'opération est terminée. Il ne vous reste plus qu'à incorporer cette nouvelle fonction à vos programmes.

ADDITIF

Le programme refuse les tableaux à deux dimensions et affiche une ERROR 1. D'ailleurs, quelque soit l'erreur commise, il affichera toujours ERROR 1, car la programmation en est plus simple. Vous pouvez travailler très facilement sur plus de 5 tableaux en modifiant la valeur en &16D pour la version 16 Ko ou &N(N+1) 6D pour une autre version. Cet octet vaut 5 dans le programme actuel, mais vous pouvez mettre une valeur supérieure. Il faut simplement savoir que la taille du programme augmente de 3 octets par variable supplémentaire. Ainsi, si vous y placez 8, le programme aura (8-5)x3 = 9 octets de plus. Il faudra donc faire un NEW &29B + 9 pour une version 16 Ko.

MARC GIRONDOT (c) 1986 M.G.

```
00C0:00 00 00 00 00 FD 1A FD    D4
00C8:1A 54 DE 25 D0 08 22 24    57
00D0:AE 02 83 C2 2C 1b DE 19    03
00D8:D0 08 16 24 AE 02 84 C2    E0
00E0:2C 0F A5 02 83 A7 02 84    72
00E8:83 07 DE 05 D0 04 02 8E    b9
00F0:01 E4 A4 bF 80 4A 02 8b    8F
00F8:01 42 04 AE 02 85 Eb 02    61
0100:86 FF 68 02 6A 89 FD A8    88
0108:C2 2C 67 55 28 55 b7 24    0b
0110:8b 04 b7 28 89 02 34 56    94
0118:b9 6F bb 80 2A 15 b7 24    96
0120:89 05 54 24 bb 20 2A FD    29
0128:A8 C2 28 4E C2 2A 4b C2    02
0130:29 48 FD 2A A5 78 9A 0A    8A
0138:A5 78 99 08 84 A7 78 64    FE
0140:8b 38 45 A6 89 35 45 26    18
0148:89 32 FD 2A 44 44 45 A7    9F
0150:02 84 81 26 45 36 89 22    A4
0158:44 84 2E 64 04 2E 64 46    8F
0160:05 2E 64 FD A8 A5 02 86    CA
0168:DD AE 02 86 b7 05 99 68    39
0170:8E 01 C6 A5 02 86 b7 FF    A9
0178:89 11 E4 44 FD A8 45 28    4D
0180:45 2A F9 02 0A A4 82 FD    18
0188:2A 9E 4F FD 2A FD 98 A5    01
0190:02 83 AE 02 87 E9 02 88    C0
0198:00 A5 02 8b b7 88 89 02    95
01A0:b5 08 FD C8 1A A5 02 87    6b
01A8:2A 34 28 18 CD 50 A5 02    0b
01b0:8A F9 02 0A A5 02 89 82    F2
01b8:08 FD 88 A5 02 8b b7 88    b7
01C0:8b 12 AE 7A 07 84 AE 7A    39
01C8:05 04 AE 7A 06 b5 D0 AE    33
01D0:7A 04 8E 05 58 7A bE F7    69
01D8:3F FD 0A FD 8A F9 02 0A    Ab
01E0:34 82 08 b5 10 AE 78 94    1E
01E8:A5 02 8b b7 88 8b 18 AE    Ab
01F0:7A 17 84 AE 7A 15 04 AE    F5
01F8:7A 16 b5 D0 AE 7A 14 A5    EF
0200:02 85 bE D0 F9 8E 0b 58    01
0208:7A bE F7 17 A5 02 85 bE    3A
0210:D0 D2 8b 54 A5 02 86 2A    EA
0218:48 02 4A 89 FD A8 FD 88    61
0220:44 44 05 b7 88 89 02 b5    2E
0228:08 1A A5 02 87 2A 34 28    00
0230:18 CD 50 FD 1A FD 98 54    67
0238:15 F9 02 0A 56 15 82 08    49
0240:54 54 15 b7 88 89 02 b5    7E
0248:08 FD C8 F9 02 1A 34 82    E2
0250:18 FD 8A DF 2A 15 F5 46    4A
0258:41 88 06 FD 0A 44 44 44    FC
0260:FD 2A 88 48 Eb 02 88 01    CF
0268:A5 02 87 DD AE 02 87 A7    53
0270:02 84 9b 03 bA 01 99 A5    7F
0278:02 88 36 8b 03 bA 01 8F    12
0280:FD 1A E2 00 32 01 01 32    E1
0288:00 6E 68 88 6b 31 10 00    94
0290:00 00 00 00 00 00 00 00    92
0298:00 00 00 00 00 00 00 00    9A
```

```
10 DATA &D1,&DD,&E3,&E6,&FC,&FF,&103,&150,&166,&16A
20 DATA &174,&190,&193,&196,&19A,&1A6,&1AF,&1B5,&1BC,&1E9
30 DATA &200,&20D,&215,&219,&22B,&265,&269,&26D,&270,&275,&278,&27E
40 INPUT "Adresse(nnC5)";A:IF (A AND &FF)<>&C5THEN 40
50 B=INT (A/256):RESTORE
60 FOR I=1TO 32:READ C:POKE B*256+C,PEEK (B*256+C)+B:NEXT I
70 BEEP 5
```

INITIATION LH-5801

Cette série avait été commencée avec le numéro 7 du dernier trimestre 83, il y a donc 2 ans. Mais elle avait commencé à un niveau trop élevé et continué sur des bases qui ont découragé à peu très tout le monde.

Pour ce premier chapitre nous n'allons pas encore plonger dans l'univers fabuleux et quelque peu magique du LM, nous n'allons pour l'instant qu'explorer le PC et sa mémoire.

Vous savez qu'un ordinateur est un engin très compliqué si on essaye de voir ce qu'il a dans le « ventre ». Nous allons donc nous limiter à quelques généralités. Tout en haut, il y a le microprocesseur, LH-5801 de son prénom, qui gère tout. C'est à lui que l'on donne les ordres, et qui transmet ces ordres à tout ce qui l'entoure. Il gère tout seul 524 280 informations. On peut en effet imaginer la mémoire comme une ligne de 524 280 cases qui peuvent, chacunes séparément, être vides ou pleines. On peut même pousser l'analogie plus loin en imaginant que sur certaines cases on a posé une plaque de plexiglas, il est alors impossible d'en modifier le contenu. Celà correspond à la mémoire morte ou ROM (MEM). Si on peut en modifier le contenu, on l'appellera mémoire vive ou RAM (MEV).

Mais pour gérer plus facilement cette succession de case, on les regroupe par 8 pour former des octets, la case s'appelant un bit. Ne confondez pas le byte anglais signifiant octet au bit français.

Donc les bits de numéro 0 à 7 formeront le premier octet, les bits 8 à 15, le second et ainsi de suite. Notre mp peut donc gérer 524280/8 = 65535 octets.

Notez que le fait de regrouper le bits par 8 n'est pas universel dans l'univers diversifié des micropocesseurs, certains les prennent par 4, (montre à quartz) et d'autres par 16 ou 32, ordinateurs professionnels. C'est pourquoi les microprocesseurs sont dits 8, 16 ou 32 bits. Le LH-5801 est donc un mp 8 bits. Mais il est tout de même un peu spécial car il peut en effet gérer 8 séries de 524280 cases, soit 8x65 535 = 52 4280 octets. On appelle Kilo-octet ou Ko un ensemble de 1024 octets. Notre mp peut donc gérer 524 280/1 024 = 512 Ko. Les compatibles IBM n'ont qu'à bien se tenir.

524 280 bits	RPU	RPV	MEØ
	SPU	RPV	MEØ
= 65 535 octets	RPU	SPV	MEØ
	SPU	SPV	MEØ
	RPU	RPV	ME1
	SPU	RPV	ME1
	RPU	SPV	ME1
	SPU	SPV	ME1

Nous allons voir comment est utilisée toute cette mémoire, comment sont stockées les informations, et comment les exploiter.

MODIFIER UNE INFORMATION

Je vous renvoie à l'article sur les fonctions logiques des ordinateurs qui devrait se trouver dans ce numéro (Il se passera 4 mois entre l'écriture de cet article et sa publication, il peut s'en passer des choses !)

Lisez au moins le paragraphe sur la représentation d'un nombre en binaire, sans quoi, ce n'est pas la peine de continuer cette lecture, vous allez être décroché.

.....

Nous revoici ensemble, vous savez à présent convertir un nombre de 0 à 255 en une succession de 8 bits. On vérifie si la leçon a été bien retenu :

128 ? 10000000 facile n'est pas, et bien continuons.

PEEK ET POKE

Nous ne vous ferons pas le classique « et CALL et gram », car le CALL n'a pas encore sa place ici, on verra dans deux mois.

Choisissez un numéro d'octet, au hasard, ce qui vous vient à l'esprit, c'est bon. Si vous avez un nombre supérieur à 65535, vous n'avez pas le droit de revenir en deuxième semaine, reprenez plutôt du début cette lecture. Vous brûlez d'impatience de connaître la valeur des 8 bits de cet octet, et bien arretez de vous consumer et faites :

PEEK (nombre) ENTER

Il s'affiche un nombre compris entre 0 et 255, il faut alors faire appel à vos souvenirs pour convertir ce nombre binaire. Pour ceux qui ont la mémoire très courte, voici un programme l'effectuant : dans A la valeur

```
10 : WAIT 0 : FOR I :7 TO 0 STEP -1 : PRINT
STR$(A)> =2 I) ; :A=-(2 I)x(A)> =2 I) :
NEXT I : WAIT : PRINT
```

Vous savez à présent comment lire et exploiter l'information lue. Pour la modifier, c'est à peine plus difficile. Il faut toutefois se rappeler que certaines cases sont couvertes de plexiglass, il ne sert à rien d'essayer de modifier ces valeurs. Nous allons travailler sur l'octet numéro 16384 car il est toujours en RAM quelle que soit votre version de PC-1500.

Faites donc pour vous entraîner :

PEEK 16384 ENTER

Il s'affiche un chiffre que vous vous empressez de transformer en binaire. Vous connaissez la position des 8 bits de cet octet.

Maintenant choisissez la valeur que vous allez y mettre, choisissez tout d'abord les valeurs des 8 bits, puis faites la conversion en décimal : par exemple

01010101 soit 85 en décimal. Pour positionner les 8 bits de l'octet 16384 à ces valeurs, il suffira de dire que l'octet 16384 prend pour valeur 85 :

POKE 16384,85 ENTER

Et comme précédemment, on vérifie que tout c'est bien passé en faisant :

PEEK 16384 ENTER

et on lit 85 !

LA MEMOIRE

Il faut maintenant déterminer où se trouve la MEV et la MEM. En théorie c'est assez simple, si vous pouvez changer la valeur d'un octet, il se touve dans une partie de MEV, sinon il se trouve dans de la MEM. Je vous épargne cette recherche le résultat ayant déjà été publié dans le numéro 12. Rappelons toutefois que :
0-32767 zone de la MEV
et 32768-65535 zone de la MEM.
Remarquez que la zone n'est pas forcément occupée, si vous n'avez aucune extension mémoire, la zone 0-32767 sera en grande partie inutilisée.

A l'intérieur de ces zones nous en distinguons encore d'autres : 0-28671 zone MEV utilisateur, elle vous appartient, vous pouvez la gérer comme bon vous semble
28672-31743 zone appelée couramment MEV système. Vous n'avez pas à y toucher lors d'une utilisation normale.
31744-32767 une zone utilisable que sur PC-1500A
Pour la MEM, reportez vous au numéro 12.

STRUCTURE DE LA MEV SYSTEME

C'est une partie de la mémoire assez spéciale où est consignée la position de la machine à tout instant. Si vous êtes en mode RUN, PRO, RESERVE, TRON, TROFF, BEEP ON ou OFF,le PC le sait en regardant un octet ou un bit précis de cette MEV.
Par exemple, faites :
BEEP OFF ENTER
PEEK 30827 ENTER
Vous obtenez un nombre qui converti en binaire donne ceci :
XXXXXXX1 où X vaut 0 ou 1
Faites alors :
BEEP ON ENTER
PEEK 30827 ENTER
et vous obtenez un nombre donnant :
XXXXXXX0 en binaire. Vous vous doutez que le PC sait s'il se trouve en mode BEEP ON ou OFF en regardant le bit 0 de l'octet 30827. Par convention et analogie avec l'écriture en LM, on notera (30827) pour désigner la valeur prise par l'octet 30827.
Si vous faites un BEEP 1, la douce mélodie du Buzzer retentit. Maintenant placez le bit 0 de (30827) à 1, faites : POKE 30827, PEEK 30827 OR 1 ENTER
Pour comprendre le rôle du OR, voir l'article sur les fonctions logiques. Faites alors :
BEEP 1 ENTER
et bien sûr rien ne se produit car vous venez de passer en BEEP OFF.

LES INDICATEURS

	bØ	b1	b2	b3	b4	b5	b6	b7
30286	BUSY	SHIFT	πJ	**SMALL**	**III**	**II**	**I**	
30287	DE	G	**RAD**		**RESERVE**	**PRO**	**RUN**	**DEF**

Vous pouvez donc de la même façon que pour le BEEP, choisir dans quel MODE vous voulez vous trouver. Si le bit prend la valeur 1, l'indicateur correspondant est allumé. Après avoir choisi en binaire la valeur de l'octet correspondant à ce que vous voulez allumer, convertissez le en décimal, en POKEZ. Pour ceux à qui la conversion en décimal pose encore des problèmes, voici une ligne BASIC le faisant :

```
10 : A=0 :FOR I=1 TO 8 : A=A+2 (8-I)xVAL MID$(A$,I,1) : NEXT I : PRINT A
```

Faites A$=« 01010101 » ENTER et RUN ENTER, il s'affiche la valeur en décimal.

L'ECRAN

Vous le trouvez certainement minuscule, pourtant sachez qu'il est d'une complexité incroyable. On en arrive même à se demander si le concepteur qui a choisi cette disposition avait bien toute sa tête.
Un shéma valant mieux qu'un long discours...

	30208-10 82-30284	30464 30540	30208 30284	30464 30540
	bØ b1 b2 b3	bØ b1 b2 b3	b4 b5 b6 b7	b4 b5 b6 b7
	bØ b1 b2	bØ b1 b2	b4 b5 b6	b4 b5 b6
	30209-11 83-30285	30465 30541	30209 30285	30465 30541
G. CURSOR :	Ø 38	39 77	78 116	117 155

Faisons tout de suite quelques remarques sur ce shéma. A part qu'il est particulièrement compliqué, nous pouvons signaler que la partie supérieure de l'écran est définie par les octets de numéro pair et la partie basse par les octets de numéro impair. De plus, les bits 3 et 7 des octets de numéro impair ne sont jamais utilisés. Rassurez vous, vous n'aurez jamais à utiliser ce shéma, nous avons déjà publié dans le numéro 11 les routines permettant d'écrire ou de dessiner sans avoir à utiliser ce shéma

LES VARIABLES FIXES

Nous n'ajouterons pas grand chose à ce qui a été publié dans le numéro 14 sur les routines de calcul, et dans le numéro 12 sur la position en mémoire des variables. Juste ce petit résumé sur le codage des nombres décimaux : soit N = a.bcdefghij E nm

OCTET 1 : si nm ⩾ 0 nm
si nm < 256-nm

OCTET 2 : si N0 00
si N<0 80
OCTET 3 : ab
OCTET 4 : cd
OCTET 5 : ef
OCTET 6 : gh
OCTET 7 : ij
OCTET 8 : 00

Nous allons arrêter ici cette étude, le prochain chapitre prenant un article à lui seul :

STRUCTURE DE LA MEV UTILISATEUR

Bien sûr j'attends tous vos commentaires, et surtout tous les points à reprendre s'ils n'ont pas été clairement expliqués.

(c) 1986 Marc GIRONDOT

GALAXY FLY

Jeu tiré d'une revue Japonaise.

Tapez le programme grâce au moniteur SHARP, la colonne de droite étant celle de la somme de contrôle, ou bien si vous ne possédez pas ce programme, tapez-le normalement : POKE, &4008, &48, &70, &4A, &4E, &58, &41, &5A, &EB ENTER
POKE &40D0, &05, &51,...

Pour lancer le jeu, faites un CALL &40C8. Votre vaisseau apparaît au bas de l'écran, vous le faites bouger avec les touches = et ≫.

Les vaisseaux à détruire apparaissent par le haut de l'écran. Utilisez SPACE pour tirer.

Comme d'habitude, le but du jeu est de ne pas se faire toucher par un projectile quelconque. Version : PC-1500 A ou PC-1500 + CE – 151 ou CE-155

```
40C8:48 70 4A 4E 58 41 5A Eb    36
40D0:05 51 b5 10 41 05 1E b5    44
40D8:00 0E bE 50 88 bE 43 40    FD
40E0:48 41 4A D0 b5 00 41 4E    07
40E8:DC 91 05 b5 03 AE 41 D2    13
40F0:AE 41 D5 b5 E7 AE 41 C2    41
40F8:b5 b5 AE 41 C3 bA 50 E0    3E
4100:00 00 02 3E 00 00 3E 2A    E9
4108:3E 00 00 00 00 00 00 00    87
4110:00 3E 7F 63 41 41 41 51    85
4118:37 76 10 00 00 40 78 7C    4A
4120:56 13 13 56 7C 78 40 00    67
4128:00 41 7F 7F 46 0C 0C 46    4C
4130:7F 7F 41 00 00 41 7F 7F    EF
4138:49 49 49 5D 41 63 00 00    55
4140:00 00 00 00 00 00 3E 7F    3E
4148:41 41 41 41 41 7F 3E 00    8b
4150:00 00 01 07 1F 39 60 60    b1
4158:39 1F 07 01 00 00 41 7F    b9
4160:7F 49 49 49 5D 41 63 00    FC
4168:00 00 41 7F 7F 49 09 19    53
4170:39 6F 46 40 00 00 00 00    DF
4178:00 00 00 00 00 00 00 02    bb
4180:3E 00 00 3E 2A 3A 00 06    A7
4188:02 3E 00 3E 22 3E 00 00    A7
4190:00 00 00 00 00 00 00 00    D1
4198:00 02 3E 00 07 08 00 01    29
41A0:00 00 00 00 00 00 00 00    E1
41A8:01 00 00 00 00 00 00 00    EA
41b0:00 00 00 00 00 00 00 00    F1
41b8:EA 00 00 00 00 00 00 00    E3
41C0:FF 00 AF 62 12 00 00 00    23
```

```
41C8:06 12 00 02 04 0b 90 00    C2
41D0:01 67 00 22 01 03 00 00    9F
41D8:15 06 2D 00 08 14 05 61    E3
41E0:04 72 4D 2E 47 4D 2E 47    1b
41E8:4D 2E 47 41 43 3C 20 53    1E
41F0:43 4F 52 45 20 54 4F 50    6D
41F8:20 33 20 3E 00 00 00 00    EA
4200:3E 22 3E 02 3E 00 3A 2A    84
4208:2E 2A 2A 3E 0E 08 3E 2E    8C
4210:2A 3A 3E 2A 3A 06 02 3E    9E
4218:3E 2A 3E 2E 2A 3E 00 00    96
4220:00 48 49 54 20 4D 49 53    50
4228:53 20 52 41 54 49 4F 00    5C
4230:43 4C 45 41 52 00 52 45    70
4238:41 44 59 21 00 03 01 05    82
4240:01 0b 07 00 04 05 07 FF    A4
4248:00 00 00 00 00 00 00 00    8A
4250:18 0C 06 13 0F 13 06 0C    03
4258:18 00 1C 06 13 0F 13 06    0F
4260:1C 00 12 19 0D 07 03 00    00
4268:00 00 00 03 06 0C 19 1E    F6
4270:19 0C 06 03 0C 1E 0b 1F    34
4278:0b 1F 0b 1E 0C 0C 1A 09    48
4280:19 09 19 09 1A 0C 00 00    2C
4288:00 02 1C 02 00 00 00 00    EA
4290:15 0A 04 1b 04 0A 15 00    33
4298:00 11 0A 00 11 00 0A 11    21
42A0:00 00 40 40 60 60 60 40    C2
42A8:40 00 03 01 06 02 FF 08    3D
42b0:0A 02 09 FF 53 54 41 47    35
42b8:45 00 40 20 10 50 30 70    9F
42C0:06 12 00 02 04 0b 90 00    bb
42C8:06 10 04 02 05 0b 90 00    C6
42D0:06 12 08 03 04 09 90 00    D2
42D8:06 10 10 03 05 0b 90 00    E3
42E0:06 10 20 03 05 0D 90 00    FD
42E8:05 12 20 03 05 10 90 00    09
42F0:05 12 40 03 07 12 90 00    35
42F8:05 10 40 03 07 14 90 00    3D
4300:48 70 4A 00 58 41 5A 00    38
4308:68 41 6A 4E 15 b9 0F AE    37
4310:41 9C 25 F1 b9 F0 Ab 41    Db
4318:9C 41 55 F1 b9 0F AE 41    35
4320:9C 65 b9 F0 Ab 41 9C 41    D6
4328:4E 4E 99 20 4C 71 8b 06    0E
4330:48 71 4A 00 9E 2A 9A 9A    72
4338:9A 9A 9A 9A 9A 9A 9A 9A    4b
4340:b5 00 48 41 4A 00 41 4E    9A
4348:9C 99 05 9A 9A 9A 9A 9A    C7
4350:b5 00 48 41 4A 00 41 4E    AA
4358:78 99 05 9A 9A 9A 9A 9A    b3
4360:bE 43 78 55 41 55 41 55    9D
4368:41 55 41 55 41 55 41 55    03
4370:41 55 41 55 41 9A 9A 9A    EE
4378:24 F9 b1 9F 48 41 4A 0b    06
4380:DD 40 40 40 40 40 40 DF    FF
4388:b7 00 99 0b 9A 9A 9A 9A    8E
4390:24 F9 b3 10 58 41 1A 15    7b
4398:9A 9A 9A 9A 9A 9A 9A 9A    Ab
43A0:b5 00 AE 41 9D A5 41 9E    A8
43A8:1A 05 D5 52 5E 00 99 06    2E
43b0:41 A5 41 9D DD b7 09 91    E5
43b8:17 9A 9A 9A 9A 9A 9A 9A    48
43C0:b5 00 AE 41 9D A5 41 9E    C8
43C8:1A 05 D9 52 5E 00 99 06    52
43D0:41 A5 41 9D DD b7 09 91    05
43D8:17 9A 9A 9A 9A 9A 9A 9A    68
43E0:58 42 5A 50 bE 43 60 bE    86
43E8:43 90 1A b5 06 F9 10 AE    8A
43F0:41 9E 04 F9 b1 08 0A bE    90
43F8:43 A0 9A 9A 9A 9A 9A 9A    bA
4400:A5 41 DA b9 10 b7 10 89    1D
4408:04 5A 50 8E 02 5A 59 58    95
4410:42 bE 43 60 9A 9A 9A 9A    5F
4418:9A 9A 9A 9A 9A 9A 9A 9A    2C
4420:58 42 5A 62 bE 43 60 9A    b5
4428:9A 9A 9A 9A 9A 9A 9A 9A    3C
4430:58 42 5A 6b bE 43 60 bE    F2
4438:43 90 D5 F9 b9 0F F9 b1    8F
4440:08 AE 41 9E 04 F9 b1 08    CF
4448:0A bE 43 C0 9A 9A 9A 9A    bF
4450:58 42 5A 6b bE 43 60 bE    12
4458:43 90 D5 F9 b9 0F b7 09    C5
4460:8b 10 1A b5 0A F9 10 AE    CF
4468:41 9E 04 F9 b1 08 0A bE    09
4470:43 A0 9A A5 41 C4 b7 00    92
4478:8b 03 bE 47 E3 9A 9A 9A    00
4480:bE 43 90 b7 05 83 05 bE    57
4488:43 E0 8E 1E b7 E0 83 05    bA
4490:bE 44 00 8E 15 b7 EA 83    9D
4498:05 bE 44 20 8E 0C b7 F4    48
44A0:83 05 bE 44 50 8E 03 bE    0D
44A8:44 30 9A 9A 9A 9A 9A 9A    FC
44b0:58 42 5A 74 bE 43 60 bE    7b
44b8:43 90 b7 E8 83 17 b7 08    C7
44C0:83 1A D5 1A b5 05 F9 10    53
```

```
44C8:AE 41 9E 04 F9 b1 08 0A    59
44D0:bE 43 A0 8E 07 58 42 5A    3E
44D8:7D bE 43 60 9A 9A 9A 9A    62
44E0:58 42 5A 86 bE 43 60 bE    bD
44E8:43 90 b7 F9 83 19 b7 F4    F6
44F0:83 28 b7 EE 83 04 b5 04    C4
44F8:8E 02 b5 02 AE 41 9E bE    CE
4500:43 78 bE 43 A0 8E 13 b7    F9
4508:FD 83 04 b5 02 8E 02 b5    CD
4510:04 AE 41 9E bE 43 78 bE    1D
4518:43 C0 9A 9A 9A 9A 9A 9A    FC
4520:A5 41 C0 b7 10 83 28 bE    3b
4528:43 7C 5A 00 05 b9 1F 41    A4
4530:50 5E 09 91 09 04 F9 b1    74
4538:08 0A 58 42 5A A1 b5 00    D9
4540:AE 41 9D 05 1b 50 41 A5    67
4548:41 9D DD b7 09 91 0F 9A    42
4550:bE 43 50 68 41 6A A0 25    bE
4558:b7 01 89 03 bE 44 E0 60    [illegible]
4560:6E b0 91 0D 6A A0 25 b7    47
4568:02 89 03 bE 44 b0 60 60    AD
4570:6E b0 91 0E 6A A0 25 b7    58
4578:03 89 03 bE 44 80 60 60    8E
4580:6E b0 91 0E 6A A0 25 b7    68
4588:04 89 03 bE 49 90 60 60    b4
4590:6E b0 91 0E bE 45 20 A5    5A
4598:41 C5 b7 00 8b 03 bE 49    2F
45A0:00 A5 41 C6 b7 00 8b 05    D8
45A8:bE 49 10 8E 03 bE 45 C9    61
45b0:bE 43 00 9A 9A 9A 9A 9A    F8
45b8:9A 9A 9A 9A 9A 9A 9A 9A    CD
45C0:4A 95 A5 41 D4 bE 47 C0    63
45C8:9A 4A 02 A5 41 C4 bE 47    A2
45D0:C0 9A 58 04 8E 02 58 02    b5
45D8:5A 00 56 5C 00 99 05 9A    61
45E0:5A 00 68 41 6A A0 65 06    9D
45E8:89 01 50 6E b0 91 09 9A    59
45F0:A5 41 C3 18 A5 41 C2 1A    b8
45F8:12 AE 41 C3 1A 12 92 DD    9C
4600:AE 41 C2 A3 41 DA 9A 9A    E9
4608:9A 9A 9A 9A 9A 9A 9A 9A    1E
4610:A5 41 C4 b7 00 8b 35 4A    C1
4618:03 bE 45 E0 14 A7 41 C8    08
4620:83 2A bE 45 F0 A7 41 C9    b7
4628:83 22 bE 45 F0 b9 0E F9    C6
4630:b3 A0 0A 48 41 05 b7 00    18
4638:89 12 b5 03 0E b5 10 F9    9D
4640:02 0A b5 01 0E A5 41 C4    00
4648:DF AE 41 C4 9A 9A 9A 9A    88
4650:A5 41 C7 b7 00 89 46 4A    13
4658:01 bE 45 E0 14 A7 41 Cb    49
4660:83 3b bE 45 F0 A7 41 CC    0b
4668:83 10 A5 41 C0 F9 b3 A0    33
4670:0A 48 41 05 b7 00 89 02    90
4678:8E 18 bE 45 F0 A7 41 CD    0C
4680:83 1b bE 45 F0 b9 0F F9    18
4688:b3 A0 0A 48 41 05 b7 00    70
4690:89 0b b5 01 0E b5 10 F9    EC
4698:02 0A b5 E9 0E 9A 9A 9A    64
46A0:48 41 4A A0 68 41 6A b0    1C
46A8:05 b7 02 89 0b 25 b7 08    24
46b0:89 1E A5 41 CE 2E 8E 18    25
46b8:b7 03 89 14 25 b7 05 89    bF
46C0:0F bE 45 F0 Ab 41 CA b7    75
46C8:E0 93 0A b7 06 91 0E 2E    15
46D0:40 40 60 60 4E b0 91 30    15
46D8:9A 9A 9A 9A 9A 9A 9A 9A    EE
46E0:58 42 5A b9 50 DF b7 00    b9
46E8:99 06 15 AE 70 4E 9A 9A    82
46F0:bE E4 2C b7 0C 89 0A A5    FF
46F8:41 C0 b7 0F 83 12 DD 8E    05
4700:0C b7 3D 89 0b A5 41 C0    81
4708:b7 00 8b 04 DF AE 41 C0    23
4710:9A b7 0A 81 04 b5 20 8E    9A
4718:03 F9 b3 30 bE 4b 10 9A    F1
4720:DD 58 41 5A FD 54 54 54    30
4728:DF b7 00 99 08 48 41 55    84
4730:41 55 41 55 41 9A 9A 9A    b2
4738:9A 9A 9A 9A 9A 9A 9A 9A    4F
4740:AE 41 9D F1 b9 0F bE 47    D1
4748:20 b5 00 41 A5 41 9D b9    E1
4750:0F bE 47 20 9A 9A 9A 9A    33
4758:9A 9A 9A 9A 9A 9A 9A 9A    6F
4760:48 41 4A D1 F9 8C 43 b5    C8
4768:00 8C 0E 4A 7b 68 41 6A    21
4770:D0 bE 47 80 bE 43 00 bE    Cb
4778:4D 20 9A 9A 9A 9A 9A 9A    C8
4780:65 FD C8 bE 47 A0 bE 47    9b
4788:40 40 FD 8A 1A 65 5E 00    b3
4790:89 03 bE 47 A0 bE 47 40    4D
4798:40 b5 00 bE 47 20 9A 9A    2D
47A0:1A bb 0F b7 0F 89 0b 14    39
47A8:bb A0 b7 A0 89 05 b5 AA    8E
47b0:8E 01 14 9A 9A 9A 9A 9A    9C
47b8:9A 9A 9A 9A 9A 9A 9A 9A    CF
47C0:5A 00 b7 0A 81 06 F9 b1    53
```

```
47C8:09 50 9E 0A AE 41 9D 14    b0
47D0:b7 00 89 02 b5 0A 48 41    A1
47D8:bE 47 20 40 A5 41 9D bE    C5
47E0:47 20 9A 68 42 6A 36 4A    bC
47E8:30 bE 4b 30 bE 43 00 9A    33
47F0:b5 00 AE 41 9D 48 41 4A    4b
47F8:00 05 1A b5 7F Fb 10 41    DE
4800:4E 75 91 0b bE 43 00 bE    66
4808:4A 34 A5 41 9D DD b7 10    F5
4810:91 20 b5 FF AE 41 C0 bE    2A
4818:48 4A b5 00 FD C8 bE 4D    77
4820:40 FD 8A DD b7 60 91 0C    C0
4828:A5 41 D2 DF AE 41 D2 b7    7F
4830:00 8b 16 bE 46 E0 b5 00    b2
4838:AE 41 C0 bE 45 50 bE 44    84
4840:73 bE 4A 30 b5 C0 AE 41    97
4848:C7 9A b5 00 AE 41 C5 AE    08
4850:41 C6 bE 43 40 bE 45 C9    AC
4858:bE 47 6b bE 45 C0 9A 9A    07
4860:b7 03 89 24 A5 41 D3 DD    A5
4868:AE 41 D3 b5 04 0E 04 F9    36
4870:b3 10 0A 05 1A b5 F8 0E    5F
4878:14 b7 E0 83 04 b5 04 8E    39
4880:02 b5 08 bE 47 60 8E 37    b1
4888:b7 02 89 33 b5 04 0E 04    10
4890:F9 b3 10 0A b5 F8 0E bE    17
4898:45 F0 b7 40 83 04 b5 10    58
48A0:8E 12 b7 80 83 04 b5 15    10
48A8:8E 0A b7 D0 83 04 b5 30    7b
48b0:8E 02 b5 50 AE 41 D8 bE    12
48b8:47 60 b5 90 AE 41 C6 9A    3b
48C0:bE E4 2C b7 20 8b 05 AE    Eb
48C8:41 C1 8E 2b A5 41 C1 b7    29
48D0:20 8b 24 b5 20 AE 41 C1    6C
48D8:A5 41 C5 b7 00 89 18 A5    C8
48E0:41 C0 F9 b3 A0 0A 48 41    08
48E8:05 b7 00 8b 05 bE 48 60    E2
48F0:8E 05 b5 FD AE 41 C5 bE    EF
48F8:45 F0 9A 9A 9A 9A 9A 9A    11
4900:A5 41 C0 bE 43 7C 04 F9    69
4908:b3 04 0A b5 0E 0b 0E 9A    88
4910:48 41 4A 00 b9 0C b7 0C    b4
4918:89 09 b5 00 41 4E 0b 99    Db
4920:05 8E 0C A5 41 D8 bE 47    Cb
4928:40 b5 00 41 bE 47 20 9A    66
4930:9A 9A 9A 9A 9A 9A 9A 9A    49
4938:9A A5 41 C0 F9 b3 A0 0A    17
4940:F9 b3 10 1A 48 41 58 41    81
4948:05 b7 01 89 0A 15 b7 F9    A6
4950:81 13 bE 47 F0 8E 38 b7    9F
4958:03 89 0A 15 b7 F4 81 05    7D
4960:bE 47 F0 8E 2A 4E A0 8b    CF
4968:17 42 52 05 b7 03 89 0A    AE
4970:15 b7 FA 81 05 bE 47 F0    FA
4978:8E 15 40 50 4E AF 8b 0F    8b
4980:40 50 05 b7 03 89 08 15    bE
4988:b7 FA 81 03 bE 47 F0 9A    95
4990:bE 43 90 b7 FD 83 04 5A    FF
4998:8F 8E 02 5A 98 58 42 bE    4A
49A0:43 60 9A 9A 9A 9A 9A 9A    28
49A8:9A 9A 9A 9A 9A 9A 9A 9A    C1
49b0:48 41 4A A0 58 41 5A b0    0F
49b8:05 b7 00 8b 08 15 DD 1E    60
49C0:b7 00 89 01 0E 50 40 4E    36
49C8:b0 91 13 48 41 4A C5 05    02
49D0:b7 00 8b 01 DD 41 4E C8    90
49D8:91 0b 4A DA 05 DD 43 05    0b
49E0:b3 00 0E 9A 9A 9A 9A 9A    EC
49E8:9A 9A 9A 9A 9A 9A 9A 9A    01
49F0:A5 41 D9 b9 03 b7 02 89    F6
49F8:2C A5 41 DA b7 00 89 25    92
4A00:4A 02 bE 45 E0 14 b7 00    44
4A08:89 1b bE 45 F0 b9 0E F9    A9
4A10:b3 A0 0A 48 41 05 b7 00    FC
4A18:89 0b b5 02 0E b5 10 F9    79
4A20:02 0A b5 01 0E 9A 9A 9A    08
4A28:9A 9A 9A 9A 9A 9A 9A 9A    42
4A30:b5 FF 8E 06 b5 20 8E 02    27
4A38:b5 30 AE 41 Db 5A 00 52    DD
4A40:5E 00 99 05 bE E4 2C AE    02
4A48:41 C1 b7 05 8b 0b A5 41    CC
4A50:Db DF AE 41 Db b7 00 99    6E
4A58:1C 9A 9A 9A 9A 9A 9A 9A    F4
4A60:bE 48 C0 bE 45 50 bE 49    CA
4A68:39 bE 49 b0 bE 46 A0 bE    04
4A70:49 F0 bE 46 50 bE 46 10    5b
4A78:bE 45 D6 9A 9A 9A 9A 9A    9D
4A80:bE 4A 60 bE 4A 60 bE 4A    A2
4A88:60 bE 46 F0 A5 41 C1 b7    84
4A90:05 8b 21 A5 41 D2 b7 00    FA
4A98:89 05 bA 4E 40 8E 15 A5    00
4AA0:41 C4 b7 00 89 0C 4A 03    88
4AA8:bE 45 E0 5E 00 89 03 bE    7D
4Ab0:50 48 9E 34 9A 9A 9A 9A    CC
4Ab8:A5 41 D4 DD AE 41 D4 FD    59
4AC0:C8 DF b9 07 DD 1A b5 08    25
4AC8:4A b8 FD CA 52 5E 00 99    24
4AD0:07 48 42 58 41 5A C8 45    Ab
4AD8:51 5E D0 91 06 FD 8A b7    76
4AE0:09 81 07 A5 41 CC D9 AE    F4
4AE8:41 CC b5 00 AE 41 D3 b5    6b
4AF0:50 AE 41 C4 48 41 4A A0    b0
4AF8:b5 00 41 4E C0 91 05 bE    9A
4b00:48 4A bE 48 36 9A 9A 9A    E7
4b08:9A 9A 9A 9A 9A 9A 9A 9A    23
4b10:F9 b1 1E 2A 58 FC 5A 9b    96
4b18:54 54 54 54 54 62 6E 00    D7
4b20:99 0A 48 41 55 41 55 41    C3
4b28:55 41 55 41 55 41 9A 9A    69
4b30:65 b7 00 8b 0A FD A8 bE    8F
4b38:4b 10 FD 2A 40 9E 0F 9A    8C
4b40:bE 43 40 4A 16 b5 2D bE    CC
4b48:4b 10 4A 4C b5 2D bE 4b    6F
4b50:10 4A 82 b5 2D bE 4b 10    72
4b58:4A 00 68 41 6A DC bE 47    E1
4b60:80 b5 23 FD CA 6E E2 91    Ab
4b68:0b 4A 1E b5 00 AE 41 9D    67
4b70:65 FD A8 bE 4b 10 FD 2A    05
4b78:40 A5 41 9D DD b7 03 91    AE
4b80:14 b5 24 FD CA 6E Eb 91    69
4b88:1E bE 43 00 9A 9A 9A 9A    5A
4b90:FD 98 A5 41 D7 0A b5 27    13
4b98:bE 4b 10 bE 43 00 bE E4    9F
4bA0:2C b7 00 99 07 b5 20 AE    F1
4bA8:41 D6 bE E4 2C AE 41 C1    88
4bb0:b7 45 8b 69 b7 20 8b 54    A1
4bb8:b7 0C 89 1C A5 41 D6 DD    04
4bC0:b7 5b 81 04 b5 20 8E 2E    33
4bC8:b7 21 89 04 b5 2E 8E 26    0F
4bD0:b7 2F 89 22 b5 41 8E 1E    4E
```

```
4bD8:b7 3D 89 2E A5 41 D6 DF    69
4bE0:b7 40 89 04 b5 2E 8E 0E    2E
4bE8:b7 2D 89 04 b5 20 8E 06    0D
4bF0:b7 20 83 02 b5 5A AE 41    95
4bF8:D6 FD C8 A5 41 D7 0A FD    A2
4C00:8A bE 4b 10 bE 43 00 bE    AE
4C08:4A 34 9E 62 FD 1A A5 41    CF
4C10:D6 51 A5 41 D7 F9 b3 06    F2
4C18:AE 41 D7 8E 02 FD 1A 9A    6b
4C20:48 41 FD 98 b5 5F 51 51    40
4C28:1E bE 4b 40 FD 1A bE 4b    Fb
4C30:90 A5 41 C1 b7 45 8b 1A    54
4C38:bE 4b 90 A5 41 C1 b7 45    C0
4C40:8b 10 bE 4b 90 A5 41 C1    67
4C48:b7 45 8b 06 bE 4A 30 bE    17
4C50:4A 30 9A 9A 9A 9A 9A 9A    b2
4C58:9A 9A 9A 9A 9A 9A 9A 9A    74
4C60:48 41 FD 5A 45 FD C8 45    Db
4C68:FD C8 45 51 05 1E FD 8A    b9
4C70:43 FD 8A 41 40 40 40 4E    D5
4C78:E4 89 01 40 FD 5A 45 FD    0b
4C80:C8 45 FD C8 45 FD C8 45    ED
4C88:51 45 51 05 1E FD 8A 43    A8
4C90:FD 8A 43 FD 8A 0E 9A 9A    6F
4C98:9A 9A 9A 9A 9A 9A 9A 9A    b4
4CA0:6A 00 48 41 FD 5A 50 50    D6
4CA8:05 17 81 08 89 08 40 50    bA
4Cb0:05 17 83 02 6A FF 9A 9A    3A
4Cb8:9A 9A 9A 9A 9A 9A 9A 9A    D4
4CC0:A5 41 D0 A7 41 E0 81 4A    55
4CC8:89 0A A5 41 D1 A7 41 E1    27
4CD0:81 40 8b 3E A5 41 D1 AE    0b
4CD8:41 E1 A5 41 D0 AE 41 E0    Cb
4CE0:4A DE bE 4C A0 6E FF 8b    F6
4CE8:06 5A E8 b5 8A 8E 1D 4A    b0
4CF0:DE bE 4C 60 4A DC bE 4C    b4
4CF8:A0 6E FF 8b 06 5A E5 b5    D6
4D00:54 8E 09 4A DC bE 4C 60    C8
4D08:5A E2 b5 1E AE 41 D7 bE    E8
4D10:4C 20 9A 9A 9A 9A 9A 9A    65
4D18:9A 9A 9A 9A 9A 9A 9A 9A    35
4D20:58 41 5A D5 A5 41 D0 17    02
4D28:81 13 b5 03 F9 9C 1E A5    19
4D30:41 D2 b7 06 83 07 DD AE    62
4D38:41 D2 bE 46 E0 9A 9A 9A    4A
4D40:bE 45 50 bE 49 b0 48 41    20
4D48:4A A0 68 41 6A b0 05 b7    FE
4D50:02 89 0E 25 b7 08 81 19    b4
4D58:b7 E0 83 15 b5 E0 2E 8E    25
4D60:10 b7 03 89 0C 25 b7 05    ED
4D68:81 07 b7 b0 83 03 b5 b0    8F
4D70:2E 40 40 60 60 4E b0 91    bA
4D78:2b bE 45 D2 9A 9A 9A 9A    2D
4D80:5A 00 b7 0A 81 06 F9 b1    19
4D88:09 50 9E 0A FD C8 14 b7    66
4D90:00 89 02 b5 0A bE 47 11    3D
4D98:40 FD 8A bE 47 11 9A 9A    F6
4DA0:bE 43 40 68 42 6A 21 4A    AD
4DA8:10 bE 4b 30 A5 41 D3 5A    51
4Db0:00 b7 08 81 06 F9 b1 07    F4
4Db8:50 9E 0A FD C8 14 4A 6A    8A
4DC0:bE 47 11 40 FD 8A b7 02    A3
4DC8:81 06 DD b7 07 81 01 DD    96
4DD0:bE 47 11 4A 7A b5 25 bE    8F
4DD8:4b 10 bE 43 00 bE 4A 30    b9
4DE0:bE 4A 30 9A 9A 9A 9A 9A    67
4DE8:DD 2A 58 50 5A EC b5 00    DF
4DF0:FD C8 54 FD 8A DD b7 0A    7b
4DF8:91 0A 62 6E 00 99 11 48    A2
4E00:41 b5 00 FD C8 55 41 FD    9C
4E08:8A DD b7 0A 91 0b 9A 9A    4E
4E10:65 b7 FF 8b 0C FD A8 bE    73
4E18:4D E8 FD 2A 40 40 F9 9E    D9
4E20:11 9A 9A 9A 9A 9A 9A 9A    b5
4E28:48 41 4A 11 A5 41 9E 1A    F8
4E30:05 D5 52 5E 00 99 06 41    E8
4E38:4E 75 91 10 9A 9A 9A 9A    52
4E40:b5 07 AE 41 9E 4A 11 68    9A
4E48:42 6A AA bE 4E 10 4A 46    98
4E50:68 42 6A AF bE 4E 10 A5    22
4E58:41 9E DF AE 41 9E b7 00    A8
4E60:8b 0b bE 4E 28 bE 43 00    79
4E68:bE 4A 34 9E 28 bE 43 00    b9
4E70:bE 4A 30 bE 4A 30 bE 4A    36
4E78:30 bE 4C C0 A5 41 Eb AE    3F
4E80:70 4E A5 41 EC AE 70 4F    Cb
4E88:9A 9A 9A 9A 9A 9A 9A 9A    A6
4E90:48 41 4A 00 b5 00 41 4E    F5
4E98:50 91 05 9A 9A 9A 9A 9A    CE
4EA0:b5 00 FD C8 A5 41 9D b9    A4
4EA8:01 b7 00 89 04 5A 6E 8E    91
4Eb0:02 5A 7b 58 51 b5 00 FD    30
4Eb8:C8 55 4C 41 81 01 0E 44    84
4EC0:FD 8A DD b7 0D 91 10 FD    D4
4EC8:8A DD b7 05 91 2C 9A 9A    2A
4ED0:b5 00 FD C8 b5 00 41 FD    8b
4ED8:8A DD b7 0C 91 0C 9A 9A    21
4EE0:FD 88 bE 4E D0 FD 0A FD    93
4EE8:88 58 51 5A 88 55 41 5E    3D
4EF0:95 91 06 bE 43 00 bE 4A    73
```

```
4EF8:38 FD 0A FD 88 58 51 5A    0D
4F00:95 55 41 5E A2 91 06 bE    CF
4F08:43 00 bE 4A 34 FD 0A bE    9b
4F10:4F D0 bE 43 00 bE 4A 38    bE
4F18:9A 9A 9A 9A 9A 9A 9A 9A    37
4F20:FD A8 b5 08 F9 20 2A b5    C9
4F28:7F 6E 00 8b 04 D9 62 9E    CC
4F30:08 48 41 4A 58 FD 88 FD    34
4F38:C8 09 41 FD 8A 4E 67 91    66
4F40:0A FD 0A FD 2A 58 51 5A    CA
4F48:A2 FD A8 55 6E 00 8b 04    30
4F50:D5 62 9E 08 0b 41 FD 2A    EF
4F58:5E b1 91 13 9A 9A 9A 9A    C2
4F60:b5 07 FD C8 2A bE 4F 20    87
4F68:bE 43 00 bE 4A 34 FD 8A    7b
4F70:DF b7 FF 99 13 9A 9A 9A    CE
4F78:48 41 4A 58 b5 00 41 4E    36
4F80:67 91 05 9A 9A 9A 9A 9A    CE
4F88:FD C8 FD 88 bE 4E E0 FD    0A
4F90:0A 40 40 40 40 FD 8A bE    2E
4F98:4b 10 40 40 40 40 9A 9A    76
4FA0:b5 01 FD C8 48 41 4A 00    3D
4FA8:05 D9 41 4E 75 91 07 bE    2F
4Fb0:43 00 bE 4A 34 FD 8A DD    E2
4Fb8:b7 07 91 1A 9A 9A 9A 9A    D8
4FC0:b5 00 FD C8 bE 4F 78 A5    b3
4FC8:41 D4 DD 4A 5A bE 4D 80    38
4FD0:FD 8A FD C8 2A bE 4F 20    C2
4FD8:bE 43 00 bE 4A 34 FD 8A    Fb
4FE0:DD b7 08 91 23 9A 9A 9A    4D
4FE8:bE 4F 60 48 40 4A C0 b5    Eb
4FF0:00 AE 41 9D FD 88 bE 4E    5C
4FF8:90 FD 0A FD 88 bE 4E A0    0F
5000:bE 43 00 bE 4A 38 FD 0A    98
5008:44 44 44 44 A5 41 9D DD    C8
5010:AE 41 9D b7 15 91 23 bE    2A
5018:4A 38 48 41 4A 0F 58 42    66
5020:5A b4 55 FD 98 bE 4F 88    FD
5028:FD 88 bE 43 00 bE 4A 38    3E
5030:FD 0A FD 1A 5E b9 91 16    5C
5038:bE 4F C0 bE 4A 30 bE 4A    95
5040:38 bE 4F A0 9A 9A 9A 9A    DD
5048:bE 43 50 bE 45 C9 68 42    5F
5050:6A b4 4A 18 bE 4b 30 A5    FE
5058:41 D4 4A 3C bE 4D 80 68    36
5060:42 6A 30 04 F9 b3 0A 0A    50
5068:bE 4b 30 bE 43 00 bE 4A    FA
5070:30 bE 4D A0 bE 43 40 bE    9A
5078:47 60 bE 4F E8 bE 4A b8    24
5080:9A 9A 9A 9A 9A 9A 9A 9A    A0
5088:bE 43 40 68 42 6A 3D 4A    b4
5090:12 bE 4E 10 bE 43 00 bE    CD
5098:4A 30 bE 4A 30 bE 43 40    Db
50A0:68 51 6A bF 4A 06 bE 4b    2b
50A8:30 bE 43 00 bE 4A 30 bE    1F
50b0:43 40 68 41 6A ED 4A 20    ED
50b8:bE 4b 30 bE 43 00 bE 4A    4A
50C0:30 bE 43 40 bE 4b 40 bE    88
50C8:4A 30 bE 4A 30 bE 43 40    0b
50D0:68 51 6A b1 4A 20 bE 4b    67
50D8:30 bE 43 00 bE 4A 30 9A    2b
50E0:bE 4A b8 bA 4A 80 00 00    74
50E8:00 00 00 00 00 00 00 00    38
50F0:00 00 00 00 00 00 00 00    40
50F8:00 00 00 00 00 00 00 00    48
5100:40 78 7C 56 13 13 56 7C    D3
5108:78 40 41 7F 7F 49 49 49    2b
5110:5D 41 63 00 3E 7F 63 41    C3
5118:41 41 51 37 76 10 41 7F    b9
5120:7F 49 09 09 1D 01 03 00    6C
5128:41 7F 7F 41 40 40 40 40    F9
5130:60 00 41 7F 7F 46 0C 0C    7E
5138:46 7F 7F 41 01 03 07 4D    66
5140:78 78 4D 07 03 01 3E 7F    96
5148:41 41 41 41 41 7F 3E 00    9b
5150:41 7F 7F 49 09 19 39 6F    F3
5158:46 40 01 07 1F 39 60 60    4F
5160:39 1F 07 01 41 41 63 36    2C
5168:1C 1C 36 63 41 41 4C 5E    b6
5170:3b 1b 2F 1b 3b 5E 4C 22    68
5178:7C 02 00 2C 5E 3F 1b 2F    5A
5180:1b 3F 5E 2C 11 3E 01 00    05
5188:00 00 49 49 2A 00 63 00    F8
5190:2A 49 49 00 00 00 00 49    E6
5198:49 00 00 41 00 00 49 49    05
51A0:00 00 10 38 7C 34 1E 17    1E
51A8:3F 75 3F 17 1E 34 7C 38    09
51b0:10 53 20 54 20 41 20 47    A0
51b8:20 45 20 20 20 31 00 66    65
51C0:6F 72 20 50 43 2D 31 35    38
51C8:30 30 20 20 20 62 79 20    D4
51D0:4D 2E 54 41 4E 4A 49 00    12
51D8:00 00 00 00 00 00 00 00    29
51E0:4D 41 53 41 48 49 4b 4F    7E
51E8:20 20 20 54 41 4E 4A 49    0F
51F0:20 31 39 38 32 2E 31 31    C5
51F8:00 00 00 00 00 00 00 00    49
```



PC 1500

A PROPOS DE L'ASSEMBLEUR

MAD (Moniteur Assembleur Desassembleur) a été publié dans le SHARPENTIER N° 10, sa notice comportant quelques imprécisions.

Voici les dernières découvertes concernant ce logiciel :
Pour obtenir effectivement la fonction ATP et non AT :
POKE &54A3, &50
Pour que SHIFT K affiche bien BII et non A(+ :
POKE &557F, &4A, &C0
POKE &55BB, &4B, &12
Pour que LDX Y et LDX U aient bien leur code :
POKE &54D5, &18
POKE &54D9, &59
POKE &54DA, &28
POKE &54DE, &55
Pour que la fonction Q nnnn fonctionne correctement :
POKE &5940, &55, &F0
POKE &55F0, &BE, &56, &DA, &FD, &5E
Et enfin quelques précisions sur le mode d'emploi :
= ENTER correspond à donc le code FD
On a la virgule par un appui sur le point.

Un appui sur une touche du mode réserve correspond à une lettre de l'alphabet hexadécimal. Pour obtenir les doubles fonctions sur ces touches, se servir de la touche SHIFT et MODE.

Le paramètre d'une instruction de branchement relatif est l'adresse. Les nombres 16 bits doivent être écris en hexadécimal sans le signe &.

Il faut laisser un espace entre la lettre définissant la fonction et le nombre. Il faut toujours appuyer deux fois sur ENTER pour valider.

On sort d'une fonction par deux appuis sur ENTER et du programme par deux appuis sur OFF lorsque l'on est au menu.

© 1986 Le club, Mr CANILLAC H., Marc Girondot

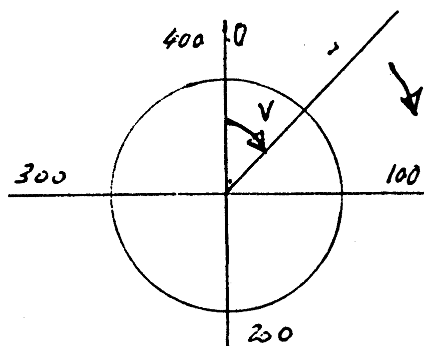
tous les PC

ASTUCES

ASTUCE 1

Suite à l'article sur les conversions REC< = >POL, voici deux astuces de nos plus attentifs lecteurs.
Les Topographes utilisent quotidiennement cette fonction mais avec des conventions géométriques différentes (v. fig. 1). Nous travaillons en Grades et comptons les angles comme indiqué sur la fig. 1.
Les formules utilisées (stockées en RSV) sont :
D : D= SQR (X*X+Y*Y) à
V : ACS (Y/D) * SGN (X+1E-4) +400* (X<0) à
Il suffit donc en mode RUN de taper
X=100-50 ENTER
Y=100-50 ENTER
puis SHIFT D la distance s'affiche : 70.71067812 m
SHIFT V le gisement s'affiche : 50 gd
On peut bien sûr adapter ces formules dans un programme, l'opération inverse étant (non programmable en RSV sur 1251) :
X=X0+D* SIN V ; Y=Y0+D* COS V

Clément GOXE.



ASTUCE 2

Dans l'article du N° 16 on préconise pour le calcul de l'angle Théta d'utiliser un diagramme puisque les ATN de 1/1, -1/1, -1/-1, 1/-1 donnent 4 angles différents et que si X=0, malgré le Y/X, il faut obtenir un résultat.

En particulier dans un calcul astronomique courant : l'azimut d'un astre, il faut trouver une solution.
L'azimut est donné par :
A=ATN ((SIN (angle horaire))/(COS (angle horaire)* SIN(latitude observateur)- TAN (déclinaison de l'astre)* COS (latitude observateur))) avec aménagement de signes possibles, suivant l'origine choisie.
Ainsi j'ai vu un jour un calcul effectué sans tenir compte du quart de cercle valable, et qui donnait pour azimut d'une planète une direction voisine du pôle nord !!!..
Dans ce calcul, le numérateur et le dénominateur de la fraction ne sont certes pas des longueurs d'abcisses et d'ordonnées, mais le calcul reste similaire à : ATN (Y/X).
Je vous indique ci-après la partie du pgm qui permet suivant les signes de X et de Y, de calculer Théta (le calcul de R et la conversion inverse sont simples).
Ex :
X=2, Y=1 A=26.56 deg
X=2, Y=-1 A=333.43
X=-2, Y=1 A=153.43
X=-2, Y=-1 A=206.56
X=0, Y=3 A=90
X=0, Y=-4 A=270
Une autre astuce concernant les 1251/55/45 mais pouvant être étendue à toute la gamme, permet d'exécuter des pgms sans taper DEF (à partir d'un pgm principal) pour des menus. Les pgms doivent être identifiés par des étiquettes se référant aux 18 touches utilisables par DEF.
10 : POKE & F83C, PEEK & F83C OR 1

P. BURC KLE

```
800:"C": INPUT "X=";X:
    INPUT "Y=";Y
810:IF X=0 AND Y>=0 THEN
    LET A=90: GOTO 880
820:IF X=0 THEN LET A=27
    0: GOTO 880
830:A= ATN (Y/X)
840:IF A<0 GOTO 860
850:IF X<0 THEN LET A=A+
    180: GOTO 880
855:GOTO 880
860:IF X>0 THEN LET A=A+
    360: GOTO 880
870:A=A+180
880:PRINT A
```

SYSTEME ET SOUS-PGMS LM

Pour contenter les possesseurs de PC 1450 et PC 2500, voici une extension de ce qui a déjà été publié pour l'ensemble des PCs ESR-H. Nous voilà donc à jour. La prochaine fois, nous débuterons une nouvelle série sans oublier personne.

J.F.V.

DEFINITION	PC 1450	PC 2500
START BASIC(PDP)	5F01,5F02	6D91,6D92
END BASIC (PFP)	5F03,5F04	6D93,6D94
VARPTR (PDV)	5F07,5F08	6D97,6D98
START (RSV-1)	/	6DE5,6DE6
79 JP n m	5F59	6DE9
n	5F5A	6DEA
m	5F5B	6DEB
flag WAIT n	5F13	6DA3
flag P.=LP.	5F12	6DA2
VARIABLES FIXES	5C30 a 5CFF	6C50 a 6D1F
RSV	/	6D20 a 6D6F

DEFINITION	PC 1350	PC 2500
DATAPTR	5F22,5F23	6DB2,6DB3
FOR-NEXTPTR Pf	5F2B	6DBB
GOSUBPTR Pf	5F2C	6DBC
Input Buffer Ptr	5F2A	6DBA
Key Code DEF	5F31	6DC1
Key Code Buffer	5F57	/
USING 1 ^	5F39	6DC9
#	5F3A	6DCA
.	5F3B	6DCB
&	5F3C	6DCC
USING 2	5F44 a 5F47	6DD4 a 6DD7
GOSUB Stack	5FAC a 5FBF	7090 a 70A3
FOR-NEXT Stack	5E06 a 5E5F	6F06 a 6F5F
String Buffer	5E60 a 5EAF	6F60 a 6FAF
Input Buffer	5EB0 a 5EFF	6FB0 a 6FFF
RND Buffer	7098 a 709F	7298 a 729F
Answer Memory	5FF8 a 5FFF	70A8 a 70AF

SOUS-PGM LM	PC 1450	PC 2500
LCD ON	05A0	052E
LCD OFF	059C	052A
STX : BA-->X	029A	02A8
DSTX : BA-1-->X	02A3	02BD
STY : BA-->Y	0295	02A3
DSTY : BA-1-->Y	02AB	02DB
BA-->(P+1,P)	0296	02A4
MVXY : Y-->X	1A14	1497
MVYX : X-->Y	1A0F	1492
PUSH X	121C	1140
POP X	1227	114B
OUT C,A	05A2	0530

SOUS-PGM LM	PC 1450	PC 2500
0 --> Xreg	0132	0168
0 --> Yreg	0126	015C
0 --> Zreg	012C	0162
Xreg <-> Yreg	0224	0235
Xreg --> Yreg	01E7	01F8
Xreg --> Zreg	0204	0215
Xreg --> Wreg	0210	0221
Yreg --> Xreg	01CB	01DC
Yreg --> Zreg	01F8	0209
Yreg --> Wreg	01B1	01C2
Zreg --> Xreg	01F3	0204
Wreg --> Xreg	01D9	01EA
Wreg --> Yreg	01BD	01CE

INITIATION AU LM ESR-H (4e)

Amis lecteurs, bonjour ! La fois précédente, nous avons vu qu'un programme avait besoin de variables pour fonctionner. Cela est vrai dans tous les langages évolués comme Basic, Pascal ou Forth. Essayez donc d'écrire un programme en Basic, sans aucune variables. Votre programme ne pourra pas faire grand chose. En effet, les variables permèttent de stocker des résultats, d'effectuer des calculs, de conserver des informations etc. C'est encore plus vrai en Langage Machine où les registres tiennent lieu de variables. A ce titre, nous avons constaté que le registre le plus usité était A, l'accumulateur, par qui passe la majorité des calculs ou traitements. En résumé, les variables sont appelées registres. C'est dans ces registres que nous garderons des informations ou des données pour de futurs calculs, ou que nous effectuerons certains traitements simples.

Pour être clair, rappelons qu'il existe des registres sur 7, 8 ou 16 bits. Par conséquent les valeurs maximales qu'ils pourront contenir seront différentes : &7F, &FF, ou &FFFF. A ce titre, nous avons parlé de, l'affectation de valeur à ces mêmes registres par les instructions de type LIn : LIA &80 correspond en basic à A=&80 etc.

Voyons maintenant le mnémonique LIDP :

LIDP nm : code &10 Load Immediate valeur dans registre DP DP : registre 16 bits,nm : valeur sur 16 bits décomposée en deux octets.

nm↦ DP où 0<= nm <= 65535

On charge donc DP avec une valeur sur 16 bits qui est généralement une adresse

Ex : LIDP &6915 donne en code, 3 octets : &10 &69 &15

Nous voyons que la valeur sur 16 bits s'écrit en code machine dans l'ordre logique Poids Fort, Poids faible.

Le mnémonique LIDL est très proche du précédent :

LIDL n : code &11 Load Immediate valeur dans Low order byte de DP (soit dans le poids faible de DP) N : est cette fois un octet sur 8 bits.

n↦DP1 où 0<= n <=255

Cette instruction est pratique et économique, puisqu'elle permet de ne s'adresser qu'au poids faible de DP. Il est bien entendu que le poids Fort ne varie pas avec cette instruction, et qu'il est égal au poids Fort n de la dernière instruction LIDP nm. (ou de la dernière instruction modifiant DP comme IX ou DYS etc.)

Ex : LIDP &6915 donne &10 &69 &15 et DP vaut &6915 puis quelques instructions plus tard : LIDL &F1 donne &11 &F1 et DP vaut &69F1. On note souvent de la manière suivante : LIDP &6915 DP←&6915

....

LIDL &F1 DP1←&F1 soit DP←&69F1

On peut aussi mettre la flèche dans l'autre sens, c'est une affaire de goût : &6915↦ DP.

En bref, le programme suivant charge &6915 dans DP, attentd deux cyles, puis charge la valeur &69F1 dans ce même registre :

Adresse	Code	Mnémonique	Opèrande	Commentaire
&6900	&10 &69 &15	LIDP	&6915	&6915↦ DP
&6903	&4D	NOPW		2 cycles wait
&6904	&11	LIDL	&F1	&F1↦ DP1
&6906	&37	RTN		On revient

LIDL est très pratique à condition d'être sûr, du Poids Fort de DP. Celà demande une certaine prudence quant à son emploi. Hormis l'effet décrit, LIDP et LIDL n'affectent aucun autre registre que DP et ne modifie pas les flags C et Z.

Nous avons déjà vu le mnémonique LIP n qui affecte au registre d'index de pile P, la valeur n sur 7 bits : en effet la valeur maximale sur 7 bits est, je le rappelle, &7F et la pile va jusqu'à

&5F puisqu'elle comporte 96 octets. On pourrait considérer l'octet &5F comme un registre, mais nous verrons qu'au delà de &3F, nous n'avons plus qu'une zone mémoire pour la pile et les ports E/S.
En clair LIP n donne n↦ P et P peut être chargé sans limitation, et adresser tous les registres ou octets de la ram interne de 96 octets.
Par contre l'instruction LP 1 fait la même opération mais en un seul octet (au lieu de 2 pour LIP et n) et sur un champ réduit. Nous allons pouvoir, grâce à LP 1, charger P avec une valeur entre 0 et &3F.
LP 1 Load la valeur 1 dans le registre P
1 : valeur sur 6 bits
1↦ P où 0<= 1 <= &3F
Cette instruction est là pour gagner du temps, et de la mémoire et permet d'adresser par P les registres de la pile. Ces registres (et ce n'est pas un hasard) vont de 0 (I) à &3F. La valeur 2 correspond à A, 3 à B etc.
Comme cette instruction fait sur un octet ce que sait faire LIP n sur deux octets, la valeur 1 n'apparaît pas explicitement. Le code &80 correspond à l'instruction LP &00, &A0 donne LP &20, comme &BF donne LP&3F. On gagne donc un octet par rapport à LIP n, ainsi que 2 cyles d'exécution.

En résumé :

&80	LP &00	= &12 &00	LIP &00	&00↦ P
&81	LP &01	= &12 &01	LIP &01	&01↦ P
&81	LP &01	= &12 &01	LIP &01	&01↦ P
&81	LP &01	= &12 &01	LIP &01	&01↦ P
&BF	LP &3F	= &12 &3F	LIP &3F	&3F↦ P

Evidemment, ça fait double emploi, mais c'est tellement pratique.
Nous avons passé en revue toutes les affectations de valeur pour tous les registres courants : en effet, certains registres ne peuvent pas être chargé par une valeur dite immédiate.
Il faut passer par l'accu A. C'est un chargement indirect par opposition au chargement direct de valeurs explicites. Nous verrons cette partie importante un peu plus tard (si vous êtes sage !). N'ayez crainte, nous y arrivons à grands pas. Le mnémonique LIA n permet de charger l'accu A avec une valeur explicite, mais il y a d'autres façons de charger A. Avec par exemple, le contenu de certains registres. LDr est une instruction de ce type, elle place dans A la valeur contenue dans le registre r :
LDr Load le contenu du registre r dans l'accu A où r est un registre 7 bits d'index de ram interne
r↦ A r est P, Q ou R.
LDP : code &30 P↦ P
LDQ : code &21 Q↦ A
LDR : code &22 R↦ A
Seul le registre A est affecté, ni le registre r ni aucun autre, ou les flags ne sont modifiés.
L'opération inverse est possible, elle consiste à placer dans P, Q ou R, la valeur contenue dans A. Ceci avec l'instruction de mnémonique STr :
STr STore le contenu de l'accu A dans le registre r où r est un registre 7 bits d'index.
A↦ r r est P, Q ou R
STP : code &30 A↦ P
STQ : code &31 A↦ Q
STR : code &32 A↦ R
Ici seul r est affecté.
Nous voyons donc, qu'au travers de A on peut charger d'autres registres ou bien récupérer une valeur contenue dans un registre précis. Car, la puissance des chargements indirectes ne s'arrête pas aux seuls registres 7 bits P, Q ou R. Tous les registres ou plus généralement, tous les octets de la ram interne, c'est-à-dire chaque octet indexé de 0 à &5F, peuvent envoyer leur contenu dans A. Il suffit d'adresser le registre ou le port E/S ou un octet de la pile, dont l'adresse varie de 0 à &5F dans la ram interne de 96 octets, par P. Et ce au moyen de l'instruction LP 1, déjà vue pour les registres de 0 à &3F, ou par LIP n pour l'ensemble des octets de la ram interne. Vous allez voir, ce n'est pas si compliqué que ça en a l'air. On veut charger dans A le contenu du registre K. On fait pointer P sur K, en lui affectant la valeur 8 qui est l'adresse du registre K dans la ram interne. K est le 9e octet à partir du début qui commence à 0 par le registre 1, puis 1 donne J, 2 donne A etc. Mais ne nous égarons pas. L'instruction LP &08 nous permet d'adresser K au moyen de P (LIP &08 convient aussi). Si on avait voulu adresser J on aurait fait LP &01 et pour adresser le port FO : LIP &5E. Maintenant, P pointe sur le registre adéquat, soit K. On veut placer le contenu de K dans A, nous allons utiliser une nouvelle instruction : LDM.
LDM : code &59 LoaD le contenu de l'adresse pointée par P dans A, M signifie (P) ; ce qui est adressé par P.
(P)↦ A les parenthèses indique que l'on prend le contenu de P.
La séquence : &88 LP &08
&59 LDM
&08↦ P pointe sur K
(P)↦ A soit K↦ A
place dans A le contenu de K.

Par contre, l'opération inverse n'est pas possible directement. Nous verrons la prochaine fois comment faire, grâce à un échange de valeurs entre A et un registre pointé par P. Or dans le cas de LDM, seul A est affecté, tandis que lors d'échange de registres, le registre de départ et celui d'arrivée sont modifiés. Les instruction vues aujourd'hui ne touchent le registre d'arivée et seulement celui là.
Pour finir, il reste un registre dont le contenu (tout comme P, Q ou R) peut être placé dans A ou vice et versa. Ce registre est le seul registre d'adresse qui nous reste à voir, j'ai nommé DP. Les adresses au contraire de P (par ex) sont sur 16 bits et font référence à la ram externe : P adresse la ram interne sur 7 bits, est-il besoin de le rappeler ?
Nous allons placer le contenu de DP, noté (DP), dans A. Pour les mnémoniques quand on parle de (DP), on écrit D, tout comme on écrit M quand on parle de (P).
LDD : code &57 LoaD le contenu du registre d'adresse DP dans l'accu A
D signifie (DP) ; ce qui est adressé par DP.
(DP)↦ A
L'opération inverse se nomme STD :
STD : code &52 STore le contenu de l'accu A dans l'adresse pointé par DP.
A ↦ (DP)
On place donc le contenu de A dans (DP).
Par exemple :
LIA &09 &09↦ A
LIDP &6915&6915↦ DP
STD A↦ (DP) soit &09↦ &6915
Ensuite la séquence :
LIDL &16 &6916↦ DP
LDD (DP)↦ A
place dans A, la valeur contenue dans l'adresse &6916. On peut connaitre cette valeur par un PEEK &6916, sous BASIC.

Les instructions vues aujourd'hui, n'affectent pas les flags C et Z. Nous verrons, lors de notre prochaine rencontre les instructions qui permettent d'incrémenter (ou de décrémenter) certains registres ; c'es-à-dire rajouter 1. Dans certains cas les flags peuvent être affectés. Mais, nous pouvons être satisfait, nous savons quand même effectuer des chargements directs et indirects de registres. Nous verrons aussi, avant les INC et les DEC, deux instructions d'échanges élémentaires : EXAM et EXAB.

Puis deux instructions qui chargent (DP) par (P) et vice et versa : MVMD et MVDM. Mais ne déflorons tout de suite, ce qui sera, je l'espère une 5e leçons appréciées de tous.

Bon LM !!

Jean-François VIGNAUD.

NAVIGATION ASTRO

Ce programme est une version simplifiée du programme de navigation astronomique publié dans le Sharpentier de décembre 1984.

Il ne prend en compte que le soleil, et ne donne que la méthode de la droite de hauteur. Les calculs ont été débarrassés de différents effets de faible importance. Malgré ces simplifications la précision reste tout a fait honorable : l'erreur maximum sur le calcul du point astronomique ne dépasse pas 0,5'.

Profitons de l'occasion pour donner des expressions plus exactes pour le programme de décembre 1984 :

06 : Q=074.0298

43 : K=133 - 0.053* T

46 : G=H/24 - P+Z/B+ (Q-A+X) R-0.000005*SIN (R*(T-5)/29.53))

G. VINCENT

```
05  J=31 , M=12 , N=80                   31/12/1980
06  I = - 1462                           Nb jours/1985,00

09  W = - 4                              décalage horaire (Antilles)

10  H = 16.2416 , S =17.02               16h 24min 16s , 17deg 02'
11  Z = 2.5                              mètres
12  F = -1                               bord inférieur
15  H = 20.40444 , T = -1461.149815

20  S = 16.9865                          √( )=dépression horizon
21  Z = 0.0536                           réfraction 0 degC       (deg.)
22
23  S = 16.9367                          0.93*Z : réfraction 20 degC

40  Obliquité écliptique , Excentricité , année sidérale
41  T périgée 1985 , longitude Périgée 85 , temps sidéral MOYEN 1985,0
42  X =-1442.2150                        mouvement moyen soleil
43  U =-1442.2527                        anomalie excentrique (KEPLER)
44  V =   -2.2906                        anomalie Vraie
45  Z =   -0.0560   L =-79.6641          précession , Longitude soleil
46  A =  -78.7570                        Ascension droite        (deg.)
47  D =  -23.0404                        Déclinaison             (deg.)
48  Y =   -2.870890                      temps sidéral MOYEN     (tours)
49  G =  125.2365                        AHvo ou GHA             (deg.)
50  S =   17.2104                        Sextant corrigé         (deg.)

100 Y =   13.5000    X = 61.0400         point estimé            (deg ' ")
101 U =   64.1699                        angle horaire local     (deg.)
102 Z =   17.2019                        hauteur astronomique    (deg.)
103 A =   60.1204                        Azimut                  (deg.)
104 A =  240.1204    K =  0.512          quadrant,0<A<360 , intercept(')
105 AZIMUT = 240.1 (deg.)                INTERCEPT = 0.5 (')
```

```
1   "N" : PRINT "NAVIGATION ASTRONOMIQUE" : DEGREE
5   N=86 : INPUT "JOUR=" ; J ,"MOIS=" ; M , "ANNEE NN=" ; N
6   I = J + 31*M - 32 - (M>2) * (3 -(INT (N/4) = N/4))
        -(M>4) -(M>6) -(M>9) -(M>11) +INT (365.25*(N-85))
9   INPUT "TEMPS LOCAL - T.U" ; W
10  INPUT "TEMPS H.MNS =" ; H ,"SEXTANT D.MNS ="; S
11  Z = 2.0 : INPUT "HAUT/EAU" ; Z
12  F = - 1 : INPUT "BORD INF = ENTER, SUP = 1 " ; F
15  H = DEG H - W : T = I + H/24 : S = DEG S
20  S = S - √(Z/1140)
21  IF S > 11 LET Z = 1 / TAN S / 61 : GOTO 23
22  Z = 0.61 / (1 + 0.4*S + S*S/33*(1-S/22))
23  S = S - 0.93*Z
40  O = 23.443 : E = 0.016715 : B = 365.256363 : R = 360
41  C = 2.1230 : P = 77.3175 : Q = 100.5915
42  X = (T-C)*R/B : Z = E*R/2/π: U = X
43  FOR K = 1 TO 4 : U = X + Z * SIN U : NEXT K
44  V = 2 * ATN ( TAN (U/2) * √((1+E)/(1-E)))
45  Z = 0.014 * T/B : L = V - P + Z
46  A = ATN (TAN L * COS O) + 180 * (COS L < 0)
47  D = ASN (SIN O * SIN L)
48  Y = H/24 + T/B + (Z+Q)/R
49  G = Y -A/R : G = (G - INT G) * R
50  S = S + 0.267 *(1 + E*COS V) *(COS S/109 - F)
100 INPUT "LAT D.MNS (N+,S-)=" ; Y, "LONG (O+,E-)=" ; X
101 V = DEG Y : U = G - DEG X
102 Z = ASN (COS V*COS D*COS U + SIN V*SIN D)
103 A = ACS ((SIN V*COS D*COS U - COS V*SIN D) / COS Z)
104 A = A * SGN SIN U + 180 : K = (S-Z) *60
105 PRINT USING "#### .#" ; " AZ =" ; A, "    IN =" ; K
120 GOTO 10
```

Gilbert VINCENT

COMPLEMENT A « FONCTION » DU N° 16

Tout d'abord voici quelques erreurs :

R=X 2+Y 2:R=SQR R
T=Y/X:Y=ATN T
et,
R=X
X=X 2+Y 2:X=SQR X
T=Y/R:Y=ATN T

Dans ce même article, M. DUBUS nous explique que l'angle trouvé peut ne pas être l'angle recherché mais son complément, pour la conversion REC-POL.

Dans le manuel d'applications du PC-1211, l'ambiguité avait été levée avec élégance :

LISTING 1

J'ai amélioré ce programme afin d'obtenir un angle en fraction de PI lorsque nous sommes dans le mode RADIAN

Exemple :

RUN «RP »
• 1 ENTER
• SQR 3 ENTER
affichage de 2. -2.PI/3

LISTING 2

De plus la ligne 101, facultative permet de mettre un résultat effectué au clavier sous forme de fraction.

Exemple :

1/3+3/4 ENTER
SHIFT F (mode DEF) pour PC-1211 ou DEF F pour PC-1350
CL pour PC-1211 ou CLS pour PC-1350
S ENTER affichage du numérateur : 13
J ENTER affichage de dénominateur : 12

Le déclenchement de l'erreur peut-être évité, mais la complexité ainsi entrainée ne parait pas justifiée.

© 1986 H.FAUCON

GESTION DE PORTEFEUILLE BOURSIER

Bien entendu ce programme s'adresse à une minorité de possesseurs de PCs, mais il nous a paru intéressant de montrer une des nombreuses applications personnelles, et surtout professionnelles qu'un PC SHARP peut effectuer.

Ce programme est écrit en pur Basic sur un PC 1251, mais peut sans difficulté (en tenant compte des particularités du 1251 ; affichage notamment) être adapté à d'autres machines. Il fonctionne avec ou sans imprimante, le choix étant effectué dans le programme à plusieurs reprises. Conformez-vous aux instructions affichées, et lisez les commentaires figurant dans le listing publié du programme. Ces commentaires font office de mode d'emploi. Les exemples illustrant chacun des 3 choix du menu vous permettront de mieux comprendre le déroulement du programme.

J'espère qu'il vous rendra autant de services qu'à moi-même.

H. CARUSO.

```
  1:REM :GESTION DE PORT
     EFEUILLE BOURSIER
  2:REM :AUTEUR:H.CARUSO
  4:"B": INPUT "IMPRIMER
      OUI(1)NON(2)";J
  5:ON J GOTO 10,20
 10:PRINT = LPRINT
 15:PRINT ""
 16:PRINT ""
 17:PRINT ""
 20:INPUT "DATE?EN 6 CHI
     F.CONTIGUS:";I$
 30:PRINT "LE "; VAL
     LEFT$ (I$,2); VAL
     MID$ (I$,3,2); VAL
     RIGHT$ (I$,2)
130:REM :NLLE OPTION IMP
     RIMER UTILE SELON ME
     NU CHOISI
140:INPUT "IMPRIMER OUI(
     1)NON(2)";J
150:ON J GOTO 155,160
155:PRINT = LPRINT
160:WAIT 150
165:CLEAR
170:PAUSE "MENU"
180:INPUT "VAL(1)PORT(2)
     TOT(3)";Y
190:ON Y GOTO 200,370,61
     0,680
200:REM :VALEUR ECRITE E
     N 5 LETTRES OU SIGNE
     S "-"
220:INPUT "VALEUR?(5L,OU
      SIGNE -)";X$
225:RESTORE
230:READ M$,N,D,T,C
240:IF LEFT$ (X$,5)=(
     LEFT$ (M$,5)) THEN
     GOTO 255
245:IF M$="FINI" PAUSE "
     INCONNU": GOTO 180
250:GOTO 230
255:PRINT ""
260:PRINT "*** VALEUR :"
     ;M$
270:PRINT "NB.ACT/OBL :"
     ;N
280:PRINT "DATE ACHAT :"
     ;D
285:USING "#######.##"
290:PRINT "PX ACH.TOT.:"
     ;T
300:LET R=T/N
310:PRINT "PX REVIENT :"
     ;R
320:PRINT "DERN.COURS :"
     ;C
330:LET E=N*C
340:PRINT "ESTIMATION :"
     ;E
350:PRINT "": USING
355:IF Y=2 GOTO 420
360:GOTO 140
370:REM :ETAT DETAILLE D
     ES VALEURS DU PORTEF
     EUILLE
375:PRINT ""
380:PRINT "*************
     **********
390:PRINT "  ETAT DU POR
     TEFEUILLE"
400:PRINT "*************
     ***********"
405:CLEAR
410:RESTORE
420:S=S+E
430:READ M$,N,D,T,C
440:IF M$="FINI" THEN
     GOTO 460
445:LET Y=2
450:GOTO 260
460:PRINT ""
470:PRINT "**** VALEUR
     TOTALE ****"
480:PRINT "": USING "###
     ####.##"
490:PRINT "      ";S
500:INPUT "COUP?TRANSACT
     ?-O(1)N(2)";K
505:ON K GOTO 506,507
506:GOSUB 760
507:READ Q
508:LET L=O+Q
510:PRINT ""
520:PRINT "         LIQUID
     ITES"
530:PRINT ""
540:PRINT "      ";L
550:W=S+L
560:PRINT ""
570:PRINT "  TOTAL DU PO
     RTEFEUILLE"
580:PRINT ""
590:PRINT "      ";W
595:PRINT ""
596:PRINT ""
597:PRINT ""
599:REM :MISE A JOUR DES
      LIQUIDITES DISPONIB
     LES L.999
600:GOTO 998
610:REM :VALEUR  TOTALE
      SANS DETAIL DU PORT
     EFEUILLE
620:RESTORE
630:S=S+E
640:READ M$,N,D,T,C
650:LET E=N*C
660:IF M$="FINI" THEN
     GOTO 460
670:GOTO 630
680:END
760:PRINT ""
770:INPUT "M.COUPONS=";P
775:PRINT "M.COUPONS =",
     P
780:INPUT "M.VENTES=";V
785:PRINT "M.VENTES  =",
     V
```

```
790:INPUT "M.ACHATS=";A
795:PRINT "M.ACHATS  =",
     A
800:LET O=P+V-A
805:RETURN
845:REM :NOTER EN DATA L
     ES VALEURS AVEC "NOM
     "(8L.OU"-"),NOMBRE,D
     ATE ACHAT(EX:02.86)
846:REM :PRIX ACH.TOT.AV
     EC FRAIS,DERNIER COU
     RS
850:DATA "FINALENS",33,0
     2.85,4852.70,177.30
851:DATA "SUCR-PITH",14,
     12.86,4258.58,385.00
852:DATA "C-G-I-P.",10,0
     8.85,6582.25,778.00
853:DATA "SCHLUMBERG",11
     ,05.85,4250.58,351.5
     0
854:DATA "ENELFI-BRET.",
     25,11.85,2428.78,154
     .00
855:DATA "CROUZET",20,11
     .85,4628.86,226.50
856:DATA "SEN-MAUB.",33,
     01.86,11520.74,385.0
     0
857:DATA "AUER-",20,01.8
     6,2945.71,124.60
858:DATA "GMC-MOULIN",20
     ,10.85,4105.52,200.0
     0
859:DATA "HOECHST",10,11
     .85,6854.25,732.00
860:DATA "SONY-",50,01.8
     6,7410.74,149.70
990:DATA "FINI",-2,-2,-2
     ,-2
998:INPUT "NOTER LIQUIDI
     TES L.999";A
999:DATA 11464.06
```

```
LE 3.3.86.

*** VALEUR :SONY-
NB.ACT/OBL :50.
DATE ACHAT :1.86
PX ACH.TOT.:     7410.74
PX REVIENT :      148.21
DERN.COURS :      149.70
ESTIMATION :     7485.00

*** VALEUR :ENELFI-
NB.ACT/OBL :25.
DATE ACHAT :11.85
PX ACH.TOT.:     2428.78
PX REVIENT :       97.15
DERN.COURS :      154.00
ESTIMATION :     3850.00
```

```
*************************
  ETAT DU PORTEFEUILLE
*************************
*** VALEUR :FINALEN
NB.ACT/OBL :33.
DATE ACHAT :2.85
PX ACH.TOT.:        0.00
PX REVIENT :        0.00
DERN.COURS :      177.30
ESTIMATION :     5850.90

*** VALEUR :SUCR-PI
NB.ACT/OBL :14.
DATE ACHAT :12.86
PX ACH.TOT.:     4258.58
PX REVIENT :      304.18
DERN.COURS :      385.00
ESTIMATION :     5390.00

*** VALEUR :C-G-I-P
NB.ACT/OBL :10.
DATE ACHAT :8.85
PX ACH.TOT.:     6582.25
PX REVIENT :      658.22
DERN.COURS :      778.00
ESTIMATION :     7780.00

*** VALEUR :SCHLUMB
NB.ACT/OBL :11.
DATE ACHAT :5.85
PX ACH.TOT.:     4250.58
PX REVIENT :      386.41
DERN.COURS :      351.50
ESTIMATION :     3866.50

*** VALEUR :ENELFI-
NB.ACT/OBL :25.
DATE ACHAT :11.85
PX ACH.TOT.:     2428.78
PX REVIENT :       97.15
DERN.COURS :      154.00
ESTIMATION :     3850.00

*** VALEUR :CROUZET
NB.ACT/OBL :20.
DATE ACHAT :11.85
PX ACH.TOT.:     4628.86
PX REVIENT :      231.44
DERN.COURS :      226.50
ESTIMATION :     4530.00

*** VALEUR :SEN-MAU
NB.ACT/OBL :33.
DATE ACHAT :1.86
PX ACH.TOT.:    11520.74
PX REVIENT :      349.11
DERN.COURS :      385.00
ESTIMATION :    12705.00

*** VALEUR :AUER-
NB.ACT/OBL :20.
DATE ACHAT :1.86
PX ACH.TOT.:     2945.71
PX REVIENT :      147.28
DERN.COURS :      124.60
ESTIMATION :     2492.00
```

```
*** VALEUR :GMC-MOU
NB.ACT/OBL :20.
DATE ACHAT :10.85
PX ACH.TOT.:     4105.52
PX REVIENT :      205.27
DERN.COURS :      200.00
ESTIMATION :     4000.00

*** VALEUR :HOECHST
NB.ACT/OBL :10.
DATE ACHAT :11.85
PX ACH.TOT.:     6854.25
PX REVIENT :      685.42
DERN.COURS :      732.00
ESTIMATION :     7320.00

*** VALEUR :SONY-
NB.ACT/OBL :50.
DATE ACHAT :1.86
PX ACH.TOT.:     7410.74
PX REVIENT :      148.21
DERN.COURS :      149.70
ESTIMATION :     7485.00


**** VALEUR  TOTALE ****

      65269.40

M.COUPONS =      1250.00
M.VENTES  =      1200.00
M.ACHATS  =      2450.00

       LIQUIDITES

        11464.06

  TOTAL DU PORTEFEUILLE

        76733.46

LE        3.00       3.00
   86.00

LE 3.3.86.

**** VALEUR  TOTALE ****

      65269.40

M.COUPONS =      1500.00
M.VENTES  =      1500.00
M.ACHATS  =      3000.00

       LIQUIDITES

        11464.06

  TOTAL DU PORTEFEUILLE

        76733.46
```

LES PHASES DE LA LUNE

Le calcul des positions de la Lune est l'un des problèmes les plus complexes de l'astronomie de position.
La lune a autour de la Terre un mouvement que l'on peut grossièrement considérer comme Képlérien, mais le Soleil le perturbe de façon importante. Pour avoir la position de la Lune avec une précision suffisante, il convient en outre de tenir compte des perturbations exercées par les planètes, perturbations dites planètaires directes (action des planètes sur la lune) ou indirectes (action de planètes sur l'orbite de la terre donc indirectement sur l'orbite de la lune autour de la terre.

En général la position de la lune est donnée par ses coordonnées écliptiques géocentriques (longitude, latitude et parallaxe horizontale équatoriale).

Par définition la phase de la lune est égale à la différence entre la longitude apparente de la lune et de la longitude apparente du soleil.
Quand cette différence = 0° c'est la Nouvelle Lune
Quand cette différence = 90° c'est le Premier Quartier
Quand cette différence = 180° c'est la Pleine Lune
Quand cette différence = 270° c'est le Dernier Quartier.
Pour obtenir de manière très précise l'instant des phases de la lune, le Bureau des Longitudes calcule les positions du soleil et de la lune en utilisant les méthodes publiées par Simon Newcomb en 1895 et E.W. Brown en 1919. Rien qu'en ce qui concerne la longitude, la théorie de Newcomb pour le soleil renferme 124 termes de perturbations et celle de E.W. Brown pour la lune en contient 800. De tels calculs dépassent largement la capacité d'un micro-ordinateur de poche.

Heureusement le calcul des quatre phases principales de la lune ne nécessite pas une méthode aussi complexe.
Dans son manuel « Astronomical formulae for calculators », Jean Meeus donne une méthode concise qui ne nécessite pas le calcul explicite des longitudes de la lune et du soleil. C'est cette méthode, exposée ci-dessous, que notre programme met en œuvre.

Les **instant des phases moyennes de la lune,** affectés de l'aberration du soleil sont donnés par :
JD = 2415020,75933 + 29,53058868 k + 0,0001178 T^2 − 0,000000155 T^3 + 0,00033 sin (166°,56 + 132°,87 T − 0°,009173 T^2) (1)

Ces instants sont exprimés en jours Julien des éphémérides. Dans la formule (1) ci-dessus, une valeur entière de k donne l'instant d'une Nouvelle Lune.

Si l'on incrémente cette valeur entière de 0,25 on obtient un Premier Quartier ;
Si l'on incrémente cette valeur entière de 0,50 on obtient une Pleine Lune ;
Si l'on incrémente cette valeur entière de 0,75 on obtient un Dernier Quartier.
Toute autre valeur de k donne un résultat sans signification. Une valeur de k négative correspond à une phase d'une lunaison antérieure à 1900.

Une **valeur approchée de k est** donnée par la formule :
k = (A − 1900) x 12,3685 (2)
A est l'année considérée prise avec les décimales correspondant au mois envisagé ; par exemple pour le mois de mars 1977, on prendra A = 1977,25.

Finalement dans la formule (1), T exprimé en siècles Julien est le temps écoulé depuis le 0 Janvier 1900 à 12 ET.
T est donné par la formule :

$$T = \frac{k}{1\,236,85} \qquad (3)$$

On calcule ensuite les angles suivants :
• **L'anomalie moyenne du soleil** à l'instant JD
M' = 359,2242 + 29,10535608 k − 0,0000333 T^2 − 0,00000347 T^3 (4)
• **L'anomalie moyenne de la lune**
M' = 306,0253 + 385,81691806 k + 0,0107306 T^2 + 0,00001236 T^3 (5)
• **L'argument de latitude de la lune**
F = 21,2964 + 390,67050646 k − 0,0016528 T^2 − 0,00000239 T^3 (6)
Ces angles exprimés en degrés décimaux seront ramenés au format 0°, 360° avant de poursuivre les calculs.
Pour obtenir **l'instant de la phase vraie,** les corrections suivantes doivent être apportées au temps de la phase moyenne calculé par la formule (1). Les coefficients des termes ci-dessous sont donnés en jours décimaux.

Corrections à apporter dans le cas d'une **Nouvelle Lune ou d'une Pleine Lune :**

```
+ (0,1734 - 0,000 393 T) sin M
+ 0,0021 sin 2M
- 0,4068 sin M'
+ 0,0161 sin 2M'
- 0,0004 sin 3M'
+ 0,0104 sin 2F
- 0,0051 sin (M + M')
- 0,0074 sin (M - M')
+ 0,0004 sin (2F + M)
- 0,0004 sin (2F - M)
- 0,0006 sin (2F + M')
+ 0,0010 sin (2F - M')
+ 0,0005 sin (M + 2M')
```

Corrections à apporter dans le cas d'un **Premier ou Dernier Quartier :**

```
+ (0,1721 - 0,0004 T) sin M
+ 0,0021 sin 2M
- 0,6280 sin M'
+ 0,0089 sin 2M'
- 0,0004 sin 3M'
+ 0,0079 sin 2F
- 0,0119 sin (M + M')
- 0,0047 sin (M - M')
+ 0,0003 sin (2F + M)
- 0,0004 sin (2F - M)
- 0,0006 sin (2F + M')
+ 0,0021 sin (2F - M')
+ 0,0003 sin (M + 2M')
+ 0,0004 sin (M - 2M')
- 0,0003 sin (2M + M')
```

En outre **dans le cas d'un Premier Quartier,** on ajoutera les termes suivants : + 0,0028 − 0,0004 cos M + 0,0003 cos M'
et **dans le cas d'un Dernier Quartier :**
− 0,0028 + 0,0004 cos M − 0,0003 cos M'

Après avoir obtenu le jour Julien JD correspondant à la phase de la lune, il reste à calculer la date calendaire et l'heure TU correspondant à ce nombre JD.

L'algorithme est le suivant :

Le jour de la semaine correspondant à une date donnée est obtenu par la formule suivante :
x = (JD + 1,5) modulo 7
Si x = 0, le jour de la semaine est un dimanche, si x = 1 c'est un lundi, ..., si x = 6 c'est un samedi.

La date est obtenue de la manière suivante :

On ajoute 0,5 à JD et soit J la partie entière et F la partie fractionnaire de ce nouveau nombre.e jour de la semaine correspondant à une date donnée est obtenu par la formule suivante :

x = (JD + 1,5) modulo 7
Si x = 0, le jour de la semaine est un dimanche, si x = 1 c'est un lundi, ..., si x = 6 c'est un samedi.
La date est obtenue de la manière suivante :
On ajoute 0,5 à JD et soit J la partie entière et F la partie fractionnairz de ce nouveau nombre.
• Si l'année Y est inférieure à 1583 (calendrier Julien), on prend A = J.
• Si l'année Y est supérieure ou égale à 1583 (calendrier Grégorien), on calcule :

$$A1 = INT\left(\frac{J}{36524,25} - 51,12264\right)$$

A = J + 1 + A1 − INT(A1/4)
Puis on calule :
B = A + 1524

$$C = INT\left(\frac{B}{365,25} - 0,3343\right)$$

D = INT (365,25C)

$$E = INT\left(\frac{B - D}{30,61}\right)$$

Le jour du mois (avec ses décimales) est alors donné par :

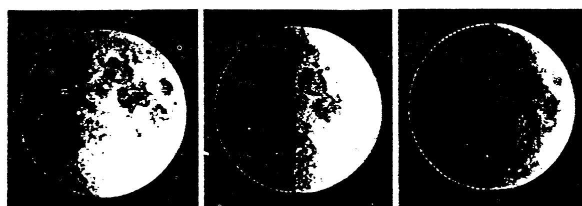
B − D − INT (30,61 E) + F

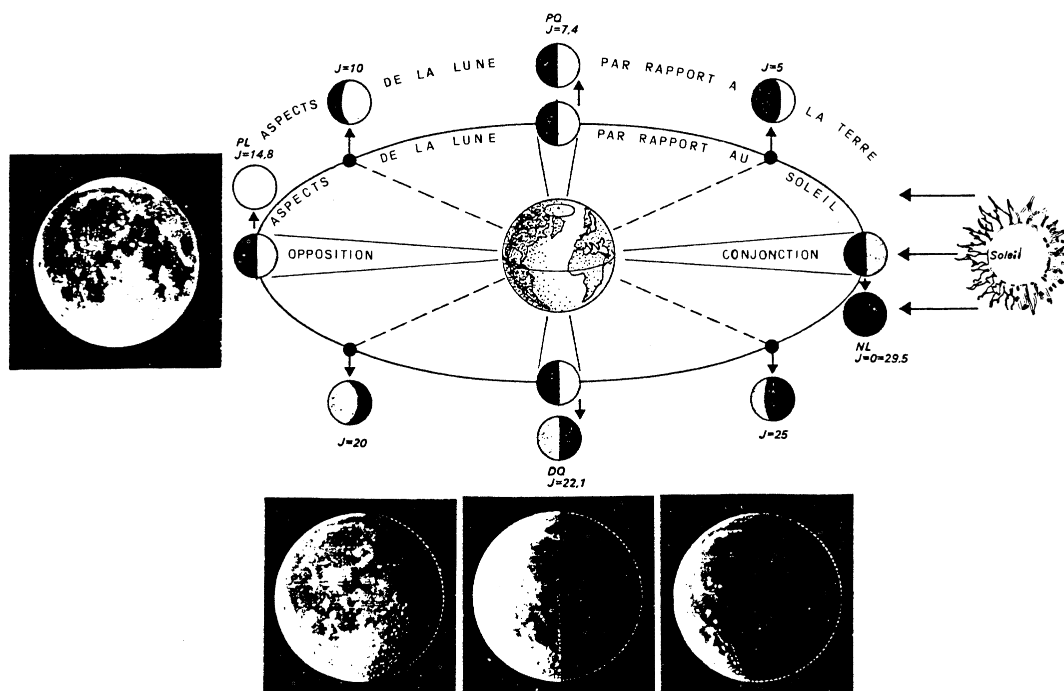
Le rang du mois est donné par :
M = E − 1 si E < 13,5 ou
M = E − 13 si E > 13,5

L'année est donnée par :
C − 4716 si M > 2,5 ou
C − 4715 si M < 2,5

On trouvera ci-après un listing commenté du programme « Phases de la Lune » écrit pour un micro-ordinnateur Sharp PC. 1261 et quelques exemples d'utilisation de ce programme. L'erreur faite sur l'heure des phases de la Lune ne dépasse pas 2 minutes de temps.

AGES ET PHASES DE LA LUNE
Tout au long de la lunaison de 29,5 jours





```
  1:"A": PRINT = LPRINT
    : GOTO 5
  2:"Z": PRINT = PRINT :
    GOTO 5
```

Entrée dans le programme par :
DEF A = avec imprimante
DEF Z = sans imprimante.

```
  3:REM PHASES DE LA LUN
    E*J. Hery d apres J.
    Meeus* Edi.20/11/85
  5:CLEAR : WAIT 100:
    DEGREE :U=0: DIM MM$
    (12)*9,L$(7)*2:
    RESTORE
```

Initialisation des variables, passage en mode degrés et déclaration des tableaux.

```
  7:FOR I=1 TO 12: READ
    MM$(I): NEXT I
 10:FOR I=1 TO 7: READ L
    $(I): NEXT I
```

Ecriture des tableaux des douzes mois de l'année et des sept jours de la semaine.

```
 20:PRINT "**Phases de l
    a LUNE**"
 25:INPUT "ANNEE ? ";Y:
    INPUT "No mois ou AN
    ? ";S$:Y1=Y
```

Entrée du millésime de l'année précédé du signe « – » si c'est une année avant J.C.

```
 30:G=1: IF Y<1583 LET G
    =0
 35:USING "#####": PRINT
    "AN:";Y: IF S$<>"AN"
    CURSOR 9: PRINT "Moi
    s: ";MM$( VAL S$)
```

Entrée du numéro du mois (ex : 3 pour mars) ou du mot AN si l'on désire les phases de la lune pour toute l'année.

Impression du millésime de l'année et du mois demandé.

```
 40:PRINT " Ph.   Date
      TU.(h.m)": CURSOR
    24: PRINT ":--:-----
    -----:--------:"
```

De 45 à 80 – Calcul de l'instant de la phase moyenne de la Nouvelle Lune la plus proche du début de l'année considérée.

JD = 2415020,75933 + 29,53058868 k + 0,0001178 T^2 – 0,000000155 T^3 + 0,00033 sin (166°,56 + 132°,87 T – 0,009173 T^2)

Pour obtenir une meilleure précision, on calcule séparément la partie entière et la partie fractionnaire de JD.

k est un entier donné par la formule : k = (Année considérée – 1900) × 12,3685

T = temps en siècles Julien écoulés depuis le 0 janvier 1900 à 12 ET

La ligne 80 convertit le temps des Ephémérides en temps universel.

```
 45:K0= INT ((Y-1900)*12
    .3685)
 50:T=(Y-1899.5)/100
 55:T2=T*T:T3=T*T*T
 60:J0=2415020+29*K0
 65:F0=.0001178*T2-.0000
    00155*T3
 70:F0=F0+.75933+.530588
    68*K0
 75:F0=F0+.00033* SIN (1
    66.56+132.87*T-.0091
    73*T2)
 80:F0=F0-.000837*T-.000
    335*T2
```

De 85 à 135 – Calcul de :

L'anomalie moyenne du soleil au temps JD = MO

L'anomalie moyenne de la lune = M1

L'argument de latitude de la lune = B1

Tous ces angles exprimés en degrés décimaux sont ramenés dans le format (0°,360°) en utilisant la fonction Modulo 360

A modulo 360 = A – 360 × INT (A/360)

$= 360 \times \left(\frac{A}{360} - \mathrm{INT}\,\frac{A}{360}\right)$

```
 85:M0=K0*.08084821133
 90:M0=360*(M0- INT M0)+
    359.2242
 95:M0=M0-.0000333*T2
100:M0=M0-.00000347*T3
105:M1=K0*.07171366128
110:M1=360*(M1- INT M1)+
    306.0253
115:M1=M1+.0107306*T2
120:M1=M1+.00001236*T3
125:B1=K0*.08519585128
130:B1=360*(B1- INT B1)+
    21.2964
135:B1=B1-.0016528*T2-.0
    0000239*T3
140:I=4*( VAL S$-1): IF
    S$="AN" LET I=0
```

```
145:FOR K9=I TO 53
150:J=J0+7*K9:F=F0+.3826
    4717*K9
155:K=K9/4
160:M5=M0+K*29.10535608
165:M6=M1+K*385.81691806
170:B6=B1+K*390.67050646
180:IF U=0 OR U=1 GOSUB
    300
185:IF U=.5 OR U=1.5
    GOSUB 340
190:F=F+.5/1440
195:J=J+ INT F:F=F- INT
    F
197:R=J+F+1.5:R=R-7* INT
    (R/7)+1
200:GOSUB 400
205:IF Y<Y1 GOTO 260
210:IF S$="AN" OR M= VAL
    S$ GOTO 220
215:GOTO 255
220:WAIT 0: CURSOR 24
225:IF U=0 PRINT :
    CURSOR 24: PRINT " N
    L";
230:IF U=.5 PRINT " PQ";
235:IF U=1 PRINT " PL";
240:IF U=1.5 PRINT " DQ"
    ;
245:PRINT "  ";L$(R);
    USING "###";D1;M;
250:USING "#####.##":
    PRINT DMS H
255:IF M> VAL S$ AND S$<
    >"AN" GOTO 270
260:U=U+.5: IF U=2 LET U
    =0
265:NEXT K9
270:PRINT : END
```

De 140 à 170 – Incrémentation de JD, MO, M1 et B1 pour les 53 phases possibles de l'année considérée. Le pas d'incrémentation est égal au quart d'une lunaison.

$\frac{29{,}53058868}{4} = 7{,}38264717$

La partie fractionnaire de K = K9/4 marque la phase :

si FRAC (K) = 0 c'est une Nouvelle Lune

si FRAC (K) = 0,25 c'est un Premier Quartier

si FRAC (K) = 0,50 c'est une Pleine Lune

si FRAC (K) = 0,75 c'est un Dernier Quartier

Dans le cas où les phases sont demandées pour un seul mois de rang m, la ligne 140 permet d'initialiser l'incrémentation de JD à la Nouvelle Lune du mois (m – 1). Le fait de soustraire 1 à m permet d'obtenir toutes les phases du mois de rang m.

La ligne 190 permet d'arrondir le résultat à la minute la plus proche en ajoutant une demi-minute à la partie fractionnaire de JD.

La ligne 197 permet d'obtenir le jour de la semaine en calculant le reste de la division par 7 du monbre JD incrémenté de 1,5.

La ligne 255 permet de sortir de la boucle d'incrémentation dans le cas où le rang du mois M calculé à partir du nombre JD devient supérieur au rang du mois demandé en ligne 25.

```
300:F=F-.4068* SIN M6
305:F=F+(.1734-.000393*T
    )* SIN M5
310:F=F+.0161* SIN (M6+M
    6)-.0004* SIN (M6+M6
    +M6)
315:F=F+.0104* SIN (B6+B
    6)+.0004* SIN (B6+B6
    +M5)
320:F=F-.0074* SIN (M5-M
    6)-.0004* SIN (B6+B6
    -M5)
325:F=F-.0051* SIN (M5+M
    6)-.0006* SIN (B6+B6
    +M6)
330:F=F+.0021* SIN (M5+M
    5)+.0005* SIN (M5+M6
    +M6)
335:F=F+.0010* SIN (B6+B
    6-M6): RETURN
```

Sous-programme des corrections à apporter à la phase moyenne pour avoir la phase vraie, dans le cas d'une nouvelle lune ou d'une pleine lune.

```
340:F=F+(.1721-.0004*T)*
    SIN M5+.0021* SIN (M
    5+M5)
345:F=F-.6280* SIN M6+.0
    089* SIN (M6+M6)
350:F=F-.0004* SIN (M6+M
    6+M6)+.0079* SIN (B6
    +B6)
355:F=F-.0119* SIN (M5+M
    6)-.0047* SIN (M5-M6
    )
360:F=F+.0003* SIN (B6+B
    6+M5)-.0004* SIN (B6
    6+M5)-.0004* SIN (B6
```

Sous-programme des corrections à apporter à la phase moyenne pour avoir la phase vraie, dans le cas d'un Premier Quartier ou d'un Dernier Quartier.

```
    +B6-M5)
365:F=F-.0006* SIN (B6+B
    6+M6)+.0021* SIN (B6
    +B6-M6)
370:F=F+.0003* SIN (M5+M
    6+M6)+.0004* SIN (M5
    -M6-M6)-.0003* SIN (
    M5+M5+M6)
380:F=F+ SGN (1-U)*(.002
    8-.0004* COS M5+.000
    3* COS M6)
385:RETURN
```

```
400:F=F+.5
405:IF F<1 GOTO 415
410:F=F-1:J=J+1
415:IF G=1 GOTO 425
420:A=J: GOTO 435
425:A1= INT ((J/36524.25
    )-51.12264)
430:A=J+1+A1- INT (A1/4)
435:B=A+1524
440:C= INT ((B/365.25)-.
    3343)
```

Sous-programme de calcul de la date calendaire et de l'heure en fonction du nombre de jours Julien JD.

```
445:D= INT (365.25*C)
450:E= INT ((B-D)/30.61)
455:D=B-D- INT (30.61*E)
    +F
460:M=E-1:Y=C-4716
465:IF E>13.5 LET M=M-12
470:IF M<2.5 LET Y=Y+1
475:D1= INT D:H=24*(D-D1
    )
480:RETURN
```

```
500:DATA "Janvier","Fevr
    ier","Mars","Avril"
510:DATA "Mai","Juin","J
    uillet","Aout"
520:DATA "Septembre","Oc
    tobre","Novembre","D
    ecembre"
530:DATA "Di","Lu","Ma",
    "Me","Je","Ve","Sa"
```

PHASES DE LA LUNE POUR L'ANNEE 1983

Phases ce la LUNE
AN: 1983

Ph	Date	TU.(h.m)	Annuaire du Bureau des Longitudes
DQ	Je 6 1	4.01	4.00
NL	Ve 14 1	5.08	5.08
PQ	Sa 22 1	5.33	5.34
PL	Ve 28 1	22.26	22.27
DQ	Ve 4 2	19.18	19.17
NL	Di 13 2	0.32	0.32
PQ	Di 20 2	17.31	17.32
PL	Di 27 2	8.57	8.58
DQ	Di 6 3	13.16	13.16
NL	Lu 14 3	17.44	17.44
PQ	Ma 22 3	2.25	2.26
PL	Lu 28 3	19.26	19.27
DQ	Ma 5 4	8.39	8.39
NL	Me 13 4	7.59	7.59
PQ	Me 20 4	8.57	8.58
PL	Me 27 4	6.30	6.31
DQ	Je 5 5	3.43	3.43
NL	Je 12 5	19.25	19.25
PQ	Je 19 5	14.17	14.17
PL	Je 26 5	18.47	18.48
DQ	Ve 3 6	21.08	21.08
NL	Sa 11 6	4.37	4.38
PQ	Ve 17 6	19.46	19.46
PL	Sa 25 6	8.32	8.32
DQ	Di 3 7	12.13	12.12
NL	Di 10 7	12.18	12.19
PQ	Di 17 7	2.50	2.51
PL	Di 24 7	23.27	23.27
DQ	Ma 2 8	0.53	0.53
NL	Lu 8 8	19.18	19.18
PQ	Lu 15 8	12.46	12.47
PL	Ma 23 8	15.00	15.00
DQ	Me 31 8	11.23	11.23
NL	Me 7 9	2.35	2.35
PQ	Me 14 9	2.23	2.24
PL	Je 22 9	6.37	6.37
DQ	Je 29 9	20.06	20.06
NL	Je 6 10	11.15	11.16
PQ	Je 13 10	19.41	19.43
PL	Ve 21 10	21.54	21.54
DQ	Sa 29 10	3.37	3.37
NL	Ve 4 11	22.21	22.21
PQ	Sa 12 11	15.48	15.50
PL	Di 20 11	12.30	12.30
DQ	Di 27 11	10.51	10.51
NL	Di 4 12	12.26	12.26
PQ	Lu 12 12	13.09	13.10
PL	Ma 20 12	2.00	2.01
DQ	Lu 26 12	18.53	18.53

PC 1261

EXEMPLE D'UTILISATION DU PROGRAMME

Petits problèmes littéraires

Les écrivains font souvent allusion à la phase de la lune dans leurs descriptions. Si la description est datée, on peut découvrir si l'auteur est fidèle aux faits.

Victor Hugo (Les Misérables) : Pleine lune à Waterloo, le **18 juin 1815**.

```
AN: 1815
Mois: Juin
 Ph.   Date      TU.(h.m)
:--:----------:--------:

 NL  Me  7  6    15.54
 PQ  Me 14  6     7.54
 PL  Me 21  6    18.02
 DQ  Je 29  6    21.46
```

Le 18 juin 1815 la lune approchait de sa plénitude. Le fait est donc correct.

Stendhal (Mémoires d'un touriste) : Clair de lune magnifique le **3 août 1837**.

```
AN: 1837
Mois: Aout
 Ph.   Date      TU.(h.m)
:--:----------:--------:

 NL  Ma  1  8    12.21
 PQ  Me  9  8    13.24
 PL  Me 16  8     5.41
 DQ  Me 23  8     1.18

 NL  Je 31  8     4.01
```

La lune était nouvelle le 1er août et donc pratiquement invisible le 3 août 1837. Souvenir inexact ou inventé.

Marcelle Tinayre (François Barbazanges) : « Ce samedi, **vingt-septième jour de juillet, l'an 1673**, naquit... La lune était vieille au dernier quartier ».

```
AN: 1673
Mois: Juillet
 Ph.   Date      TU.(h.m)
:--:----------:--------:
 DQ  Ve  7  7     6.51

 NL  Ve 14  7     1.52
 PQ  Je 20  7    18.56
 PL  Ve 28  7    19.06
```

Le 27 juillet 1673 fut un **jeudi** et la lune n'était pas encore pleine. Souvenir certainement inventé.

Jean Hery

PC 1401

CALCUL DE RESSORTS

Ecrit sur PC 1401/02 + CE 126P, ce programme fonctionne sur tous les PCs, moyennant un aménagement des affichages. Si vous ne possédez pas d'imprimante, remplacez les LPRINT par des PRINT. Nous tenons de plus, à féliciter l'auteur Mr MENARD, pour la qualité rare de sa documentation.

Ce programme sert à calculer les ressorts utilisés dans les petits mécanismes. On peut aussi bien calculer des ressorts de traction, compression ou flexion que les barres de torsion. Cette dernière possibilité n'est possible qu'avec l'extension du programme, fournie par 1402. Car la capacité mémoire du PC 1401 ne permet pas d'inclure cette option peu usitée, il est vrai : le programme occupe 3520 octets.

Il y a deux procédés de calcul : automatique et manuel.

AUTOMATIQUE

L'ordinateur recherche le diamètre de fil adéquat, en fonction des données géométrique, et de charge initiales. Cette recherche s'effectue dans deux nuances de matières : CP1 et CP2, et pour des diamètres de fils allant de 0.4 à 5mm, par essais successifs de 0.1. L'ordinateur fournit ensuite toutes les caractéristiques du ressort retenu. Ainsi que l'évaluation de sa durée de vie pour les ressorts tractions/compressions et barre de torsion.

MANUEL

Le PC fournit toutes les caractéristiques mécaniques d'un ressort défini sur plan. Dans ce mode de calcul, des changements successifs de paramètres géométriques permettent un calcul à « tatons » du ressort voulu. De plus, cela permet de dépasser les limites de travail fixées à l'ordinateur. Limites qui sont : 35% limite élastique pour les ressorts de traction/compression, 85% pour les ressorts flexion et 50% pour les barres de torsion. Dans ce mode, la machine interroge : CP1 ou CP2 ? répondre 1 ou 2 suivant que l'on désire de l'acier CP1 ou CP2. La durée de vie des ressorts hélicoïdaux est également donnée dans ce mode.

Les résultats tombent automatiquement sur l'imprimante : voir les exemples joints.

UTILISATION

RUN ou DEF K, lance le progrmme. Apparition du titre : CALCUL RESSORT. Appuyez sur ENTER : UNITES=MM/DAN. Un nouvel appui vous permet de répondre à la question : TYPE RESSORT ? par :

H : ressorts hélicoïdaux (traction/compression)

F : ressorts de flexion (cor de chasse)

B : barres de torsion (si vous êtes muni de cette option)

Validez par ENTER, toute réponse autre redonnera systématiquement cette question. Celà est vrai de manière générale pour les questions clefs.

Si vous choisissez la procédure H. La question TRA/COM ? vous est posée, répondez par :

T : pour ressort traction

C : pour ressort compression.

CALCUL ? invite à choisir le mode de calcul :

A : pour calcul automatique (détermine le diamètre du fil)

M : pour calcul manuel (on impose le diamètre du fil)

Cette question est d'ailleurs posée quelque soit l'option choisie H, F ou B.

Saisissez par la suite les données géométriques choisies. Dans le cas du calcul manuel, il faut donner le type d'acier désiré : CP1/CP2 ? (1/2) en tapant 1 ou 2, suivi du sempiternel appui sur ENTER après une entrée.

Les exemples donnés pour différents cas de figure, sont la pour vous aider à comprendre le déroulement, à la fois simple et logique du programme. Les dépassements de limites sont indiqués, et la machine demande si besoin est de refaire les calculs en CP2. La longueur hors-tout est la longueur de l'enroulement théorique, plus deux spires pour accrochage. En calcul manuel, toujours vérifier que le taux de travail à la charge (TX MAX) ne dépasse jamais la limite de déformation élastique, de même prendre une garde confortable.

Pour reprendre un autre calcul, taper 0 à la question AUTRE ESSAI ? Si vous répondez N, la machine vous adresse un rapide AU REVOIR. Bon ressorts !!!

Michel MENARD

ABREVIATIONS UTILISEES :

CH.INI : charge à la tension initiale du ressort (tract/comp)
CH.MAX : charge à la tension maximum du ressort (tract/comp)
CO.UTI : course utile = course de travail
DI.MOY : diamètre moyen d'enroulement du ressort
DI.FIL : diamètre du fil
FL.INI : flèche à la tension initiale
FL.TOT : flèche à la tension maxi
TX.INI : taux de travail à la tension initiale
TX.MAX : taux de travail à la tension maxi
NB.SPI : nombre de spire d'enroulement
CP.INI : couple à la tension initiale (flexion/torsion)
AN.TRA : angle de travail
LG.BAR : longueur barre (torsion)
PA.MIN : pas minimum (compression)
LG.ENR : longueur enroulement (traction)
LG.H.T : longueur hors tout
LG.T.I : longueur à la tension initiale
LG.DEV : longueur développée
AN.INI : angle à la tension initiale
AN.MAX : angle à la tension maxi
CP.MAX : couple à la tension maxi
DI.E.L : diamètre exterieur à l'état libre (flexion)
DI.E.I : diamètre exterieur à la tension initiale
DI.I.M : diamètre interieur à la tension maxi

```
I---------------------I
I    COMPRESSION      I
I---------------------I
I CH.INI=? I      1.0 I
I CH.MAX=? I      2.0 I
I CO.UTI=? I     10.0 I
I DI.MOY=? I     10.0 I
I DI.FIL=? I      0.9 I
I---------------------I
I      ACIER CP1      I
I---------------------I
I FL.INI=  I     10.0 I
I FL.TOT=  I     20.0 I
I TX.INI=  I     39.1 I
I TX.MAX=  I     78.2 I
I NB.SPI=  I      6.4 I
I PA.MIN=  I      4.5 I
I LG.H.T=  I     29.9 I
I LG.T.I=  I     19.9 I
I---------------------I
I DU.VIE > 10E5       I
I---------------------I
```

```
I---------------------I
I       FLEXION       I
I---------------------I
I      ESSAI CP1      I
I---------------------I
I CP.INI=? I     10.0 I
I AN.TRA=? I     45.0 I
I NB.SPI=? I      4.0 I
I DI.MOY=? I     10.0 I
I---------------------I
I DI.FIL=  I      1.0 I
I LG.DEV=  I    125.6 I
I AN.INI=  I     73.3 I
I AN.MAX=  I    118.3 I
I TX.INI=  I    101.8 I
I TX.MAX=  I    164.3 I
I CP.MAX=  I     16.1 I
I DI.E.L=  I     11.0 I
I DI.E.I=  I     11.0 I
I DI.I.M=  I      8.1 I
I---------------------I
I---------------------I
I       FLEXION       I
I---------------------I
I CP.INI=? I     10.0 I
I AN.TRA=? I     45.0 I
I NB.SPI=? I      4.0 I
I DI.MOY=? I     10.0 I
I DI.FIL=? I     10.0 I
I---------------------I
I D>DM:4 !            I
I---------------------I
I     HORS LIMITE     I
I---------------------I
I---------------------I
I       FLEXION       I
I---------------------I
I CP.INI=? I     10.0 I
I AN.TRA=? I     45.0 I
I NB.SPI=? I      4.0 I
I DI.MOY=? I     10.0 I
I DI.FIL=? I      1.0 I
I---------------------I
I      ACIER CP1      I
I---------------------I
I LG.DEV=  I    125.6 I
I AN.INI=  I     73.3 I
I AN.MAX=  I    118.3 I
I TX.INI=  I    101.8 I
I TX.MAX=  I    164.3 I
I CP.MAX=  I     16.1 I
I DI.E.L=  I     11.0 I
I DI.E.I=  I     10.9 I
I DI.I.M=  I      8.1 I
I---------------------I
```

```
I---------------------I
I      TRACTION       I
I---------------------I
I      ESSAI CP1      I
I---------------------I
I CH.INI=? I      1.0 I
I CH.MAX=? I      2.0 I
I CO.UTI=? I     10.0 I
I DI.MOY=? I     10.0 I
I---------------------I
I DI.FIL=  I      0.9 I
I FL.INI=  I     10.0 I
I FL.TOT=  I     20.0 I
I TX.INI=  I     39.1 I
I TX.MAX=  I     78.2 I
I NB.SPI=  I      6.4 I
I LG.ENR=  I      5.7 I
I LG.H.T=  I     24.8 I
I LG.T.I=  I     34.8 I
I---------------------I
I DU.VIE > 10E5       I
I---------------------I
```

```
  10:REM :M.MENARD:19.3.8
     6.
  20:"K":PRINT "CALCUL RE
     SSORT":PRINT "UNITES
     =MM/DAN"
  80:INPUT "TYPE RESSORT
     ?";O$:O$=LEFT$ (O$,1
     )
  90:IF O$<>"H" AND O$<>"
     F" GOTO 80
 100:IF O$="H" GOTO 1000
 120:IF O$="F" GOTO 3000
 1000:CLEAR :GOSUB 7000:
      INPUT "TRA/COM ?";
      R$
 1005:R$=LEFT$ (R$,1)
 1010:IF R$<>"T" AND R$<
      >"C" GOTO 1000
 1012:IF R$="T" GOSUB 96
      00:LPRINT "I      T
      RACTION        I":
      GOSUB 9600
 1013:IF R$="C" GOSUB 96
      00:LPRINT "I     CO
      MPRESSION      I":
      GOSUB 9600
 1015:GOSUB 7500:IF O$="
      A" GOTO 1040
 1030:IF O$="M" GOTO 116
      0
 1040:GOSUB 9500:GOSUB 7
      010:GOSUB 7040:
      GOSUB 9600
```

```
1050:FOR D=.4 TO 6 STEP
     .1:GOSUB 8000
1070:GOSUB 8210:Z=H*.45
     :GOSUB 9150:IF T-Z
     >1.5 NEXT D
1090:IF D>=6.1 GOTO 110
     0
1092:GOSUB 9580
1095:GOTO 1190
1100:GOSUB 9510:GOSUB 7
     000:FOR D=.4 TO 6
     STEP .1:GOSUB 8000
1130:GOSUB 8220:Z=H*.45
     :GOSUB 9150:IF T-Z
     >1.5 NEXT D
1150:GOSUB 8250
1152:GOTO 1092
1160:GOSUB 7010:GOSUB 7
     040:GOSUB 7080:
     GOSUB 7530:GOSUB 8
     000
1165:GOSUB 8260
1170:GOSUB 8270
1180:GOTO 1192
1190:GOSUB 9170
1192:IF R$="T" GOTO 120
     0
1194:IF R$="C" GOTO 220
     0
1200:GOSUB 9010:GOSUB 8
     060:GOSUB 9070
1210;GOSUB 9600:GOSUB 9
     540
1220:GOTO 9526
2200:GOSUB 9010:GOSUB 8
     080:GOSUB 9060:
     GOSUB 9072:GOTO 12
     10
3000:CLEAR :GOSUB 7000:
     GOSUB 9600:LPRINT
     "I       FLEXION
        I":GOSUB 9600:
     GOSUB 7500:IF O$="
     A" GOTO 3040
3030:IF O$="M" GOTO 316
     0
3040:GOSUB 9500:GOSUB 7
     050:GOSUB 7070:
     GOSUB 7040:GOSUB 9
     600
3050:FOR D=.4 TO 6 STEP
     .1:GOSUB 8100:
     GOSUB 8130:GOSUB 8
     150
3070:GOSUB 8210:Z=H*.85
     :GOSUB 9150:IF T-Z
     >1.5 NEXT D
3090:IF D>=6.1 GOTO 310
     0
3092:GOSUB 9580
3095:GOTO 3190
3100:GOSUB 9510:GOSUB 7
     000:FOR D=.4 TO 6
     STEP .1:GOSUB 8100
     :GOSUB 8130:GOSUB
     8150
3130:GOSUB 8220:Z=H*.85
     :GOSUB 9150:IF T-Z
     >1.5 NEXT D
3150:GOSUB 8250
3152:GOTO 3092
3160:GOSUB 7050:GOSUB 7
     070:GOSUB 7040:
     GOSUB 7080
3165:GOSUB 7530
3166:GOSUB 8260
3167:GOSUB 8270
3170:GOSUB 8100:GOSUB 8
     130:GOSUB 8150:
     GOTO 3200
3190:GOSUB 9170
3200:GOSUB 8160:GOSUB 9
     080:GOSUB 9230:
     GOSUB 9110:GOSUB 9
     120:GOSUB 9600
3220:GOTO 9526
7000:USING "######.#":
     RETURN
7010:INPUT "CH.INI=?";P
     :LPRINT "I CH.INI=
     ? I";P;" I"
7012:IF P=0 GOSUB 9490:
     GOTO 7010
7020:INPUT "CH.MAX=?";X
     :LPRINT "I CH.MAX=
     ? I";X;" I"
7022:IF X=0 GOSUB 9490:
     GOTO 7020
7030:INPUT "CO.UTI=?";C
     :LPRINT "I CO.UTI=
     ? I";C;" I"
7032:IF C=0 GOSUB 9490:
     GOTO 7030
7035:RETURN
7040:INPUT "DI.MOY=?";A
     :LPRINT "I DI.MOY=
     ? I";A;" I"
7042:IF A=0 GOSUB 9490:
     GOTO 7040
7045:RETURN
7050:INPUT "CP.INI=?";C
     :LPRINT "I CP.INI=
     ? I";C;" I"
7052:IF C=0 GOSUB 9490:
     GOTO 7050
7060:INPUT "AN.TRA=?";U
     :LPRINT "I AN.TRA=
     ? I";U;" I"
7062:IF U=0 GOSUB 9490:
     GOTO 7060
7065:RETURN
7070:INPUT "NB.SPI=?";N
     :LPRINT "I NB.SPI=
     ? I";N;" I"
7072:IF N=0 GOSUB 9490:
     GOTO 7070
7075:RETURN
7080:INPUT "DI.FIL=?";D
     :LPRINT "I DI.FIL=
     ? I";D;" I":GOSUB
     9600
7082:IF D=0 GOSUB 9490:
     GOTO 7080
7083:GOSUB 9580
7085:RETURN
7500:INPUT "CALCUL ?";O
     $:O$=LEFT$ (O$,1)
7510:IF O$<>"A" AND O$<
     >"M" GOTO 7500
7520:RETURN
7530:INPUT "CP1/CP2 ?(1
     /2)";W
7532:IF W<>1 AND W<>2
     GOTO 7530
7540:RETURN
7550:INPUT "AUTRE ESSAI
      ?";B$:B$=LEFT$ (B
     $,1)
7560:IF B$<>"O" AND B$<
     >"N" GOTO 7550
7570:RETURN
8000:K=1+((1.25*D)/A)+(
     .875*SQU (D/A))
8010:I=((8*P*A)/(π*D^3)
     )*K:F=(X*C)/(X-P):
     G=F-C:T=((8*X*A)/(
     π*D^3))*K
8050:N=(F*7850*D^4)/(8*
     X*A^3):RETURN
8060:E=D*N:J=E+(A*2)-D:
     L=J+G:RETURN
8080:M=(F+((D+.5)*N))/N
     :J=(M*N)+D:L=J-G:
     RETURN
8100:L=A*π*N:F=(11520*C
     *L)/(SQU π*20000*D
     ^4):I=(32*C)/(π*D^
     3):RETURN
8130:M=F+U:E=(C*M)/F:
     RETURN
8150:T=(E*32)/(π*D^3):
     RETURN
8160:R=A+D:X=((A*(360-(
     W/N)))/360)+D:P=((
     A*(360-(M/N)))/360
     )-D:RETURN
8190:W=(5760*C*L)/(SQU
     π*7850*D^4):I=(16*
     C)/(π*D^3):RETURN
8200:T=(E*16)/(π*D^3):
     RETURN
8210:H=233.1-33.72*D-1.
     59*SQU D+3.22*D^3-
     .68*D^4+.04*D^5:
     RETURN
8220:H=258.1-33.72*D-1.
     59*SQU D+3.22*D^3-
     .68*D^4+.04*D^5:
     RETURN
8250:IF D>=6.1 GOSUB 95
     20:GOTO 9530
8252:RETURN
8260:IF W=1 GOSUB 8210:
     LPRINT "I      ACIE
     R CP1       I":
     GOSUB 9600
```

```
8265:RETURN
8270:IF W=2 GOSUB 8220:
     LPRINT "I     ACIE
     R CP2       I":
     GOSUB 9600
8275:RETURN
9010:LPRINT "I FL.INI=
      I";G;" I":LPRINT
     "I FL.TOT=  I";F;"
      I":GOSUB 9230
9020:LPRINT "I NB.SPI=
      I";N;" I":RETURN
9060:LPRINT "I PA.MIN=
      I";M;" I":RETURN
9070:LPRINT "I LG.ENR=
      I";E;" I"
9072:LPRINT "I LG.H.T=
      I";J;" I"
9075:LPRINT "I LG.T.I=
      I";L;" I":RETURN
9080:LPRINT "I LG.DEV=
      I";L;" I":LPRINT
     "I AN.INI=  I";F;"
      I"
9090:LPRINT "I AN.MAX=
      I";M;" I":RETURN
9110:LPRINT "I CP.MAX=
      I";E;" I":RETURN
9120:LPRINT "I DI.E.L=
      I";R;" I":LPRINT
     "I DI.E.I=  I";X;"
      I"
9130:LPRINT "I DI.I.M=
      I";P;" I":RETURN
9150:USING "##.#":PAUSE
     "ESSAI EN ";D;" MM
     "
9160:USING "######":
     PAUSE "T";T;"  Z";
     Z:GOSUB 7000:
     RETURN
9170:BEEP 1:LPRINT "I D
     I.FIL=  I";D;" I":
     RETURN
9230:LPRINT "I TX.INI=
      I";I;" I":LPRINT
     "I TX.MAX=  I";T;"
      I":RETURN
9490:PRINT "MANQUE DONN
     EE":RETURN
9500:LPRINT "I     ESSA
     I CP1       I":
     GOSUB 9600:RETURN
9510:LPRINT "I     ESSA
     I CP2       I":
     GOSUB 9600:RETURN
9520:LPRINT "I    HORS
     LIMITE      I":
     GOSUB 9600:RETURN
9526:GOSUB 7550:IF B$="
     O" GOTO 80
9528:IF B$="N" GOTO 953
     0
9530:PAUSE "AU REVOIR":
     END
```

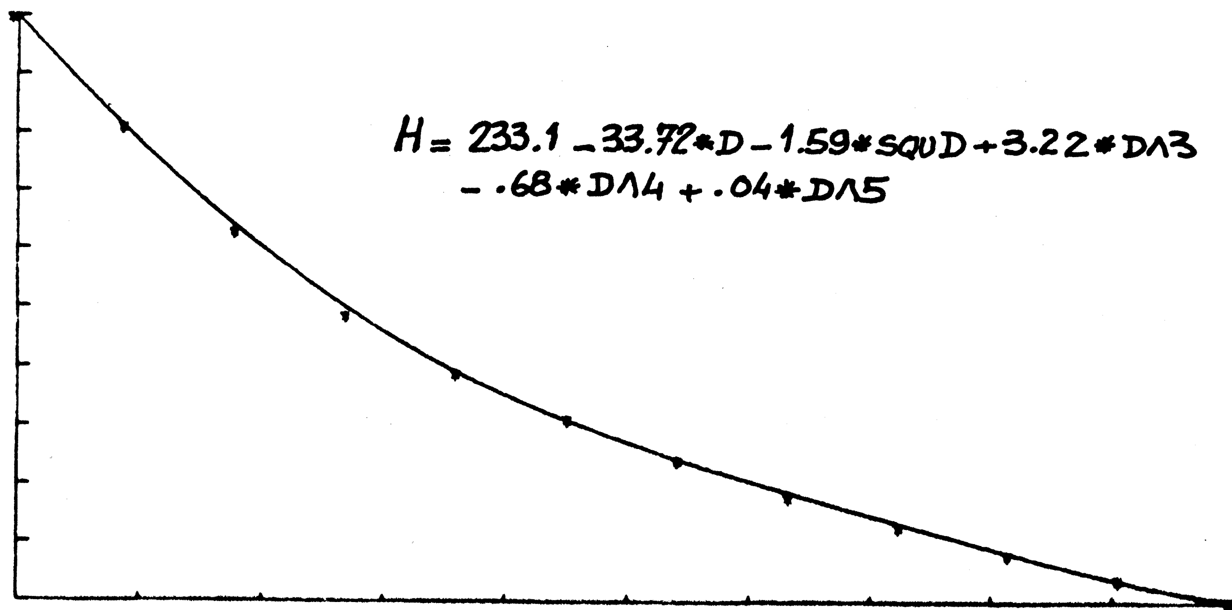
```
9540:USING "###.#^":Q=.
     7-(T/H):S=.7-(I/H)
     :V=Q/S:V=ATN V:Y=(
     V-21.5)/(6.1/10^5)
9545:IF Y<=0 GOSUB 9520
     :GOTO 9530
9550:IF Y>10^5 LPRINT "
     I DU.VIE > 10E5
        I":GOSUB 9600:
     RETURN
9560:LPRINT "I DU.VIE=
        ";Y;" I":GOSUB 9
     600:RETURN
9580:IF D>=A/4 LPRINT "
     I D>DM:4 !
        I":GOSUB 9600:
     GOSUB 9520:GOTO 95
     30
9585:RETURN
9600:LPRINT "I---------
     -----------I":
     RETURN
```

```
140:IF O$="B" GOTO 4000
 4000:CLEAR :GOSUB 7000:
      GOSUB 9600:LPRINT
      "I      BARRE TORSI
      ON    I":GOSUB 9600
      :GOSUB 7500:IF O$=
      "A" GOTO 4040
 4030:IF O$="M" GOTO 416
      0
 4040:GOSUB 9500:GOSUB 7
      050:GOSUB 8230:
      GOSUB 9600
 4050:FOR D=.4 TO 6 STEP
      .1:GOSUB 8240
 4070:GOSUB 8210:Z=H*.45
      :GOSUB 9150:IF T-Z
      >1.5 NEXT D
 4090:IF D>=6.1 GOTO 410
      0
 4092:GOSUB 9580
 4095:GOTO 4190
 4100:GOSUB 9510:GOSUB 7
      000:FOR D=.4 TO 6
      STEP .1:GOSUB 8240
 4130:GOSUB 8220:Z=H*.45
      :GOSUB 9150:IF T-Z
      >1.5 NEXT D
 4150:GOSUB 8250
 4152:GOTO 4092
 4190:GOSUB 9170
 4192:GOSUB 9085:GOSUB 9
      110:GOSUB 9230:
      GOSUB 9600:GOSUB 9
      540
 4200:GOTO 9526
 8230:INPUT "LG.BAR=?";L
      :LPRINT "I LG.BAR=
      ? I";L;" I"
 8232:IF L=0 GOSUB 9490:
      GOTO 8230
 8235:RETURN
 8240:GOSUB 8190:GOSUB 8
      130:GOSUB 8200:
      RETURN
 9085:LPRINT "I AN.INI=
       I";W;" I"
 9090:LPRINT "I AN.MAX=
       I";M;" I":RETURN
```

D=0.4	H=219.55	D=2.3	H=170.21	D=4.2	H=149.10
D=0.5	H=216.21	D=2.4	H=168.59	D=4.3	H=148.35
D=0.6	H=212.91	D=2.5	H=167.05	D=4.4	H=147.51
D=0.7	H=209.67	D=2.6	H=165.50	D=4.5	H=147.68
D=0.8	H=206.08	D=2.7	H=164.21	D=4.6	H=145.84
D=0.0	H=203.40	D=2.8	H=162.90	D=4.7	H=145.01
D=1.0	H=200.39	D=2.9	H=161.56	D=4.8	H=144.17
D=1.1	H=197.48	D=3.0	H=160.48	D=4.9	H=143.33
D=1.2	H=194.82	D=3.1	H=159.36	D=5.0	H=142.52
D=1.3	H=191.89	D=3.2	H=158.28	D=5.1	H=141.71
D=1.4	H=189.25	D=3.3	H=157.25	D=5.2	H=140.91
D=1.5	H=186.72	D=3.4	H=156.26	D=5.3	H=130.14
D=1.6	H=184.30	D=3.5	H=155.21	D=5.4	H=139.40
D=1.7	H=181.98	D=3.6	H=154.58	D=5.5	H=138.80
D=1.8	H=179.76	D=3.7	H=153.48	D=5.6	H=136.04
D=1.9	H=177.65	D=3.8	H=152.60	D=5.7	H=137.44
D=2.0	H=175.63	D=3.9	H=151.72	D=5.8	H=136.01
D=2.1	H=173.74	D=4.0	H=150.08	D=5.9	H=136.47
D=2.2	H=171.93	D=4.1	H=150.03	D=6.0	H=136.13

IMPRESSION DES VALEURS DU POLYNOME

```
B( 0)=233.10
B( 1)=-33.72
B( 2)= -1.59
B( 3)=  3.22
B( 4)= -0.68
B( 5)=  0.04
```



MODE ANGULAIRE

Grignoteurs d'octets à vos PC : il est possible d'économiser sept octets dans l'excellente astuce de monsieur GIRONDOT parue sur le quatorzième bulletin permettant de connaître dans quel mode angulaire se trouve votre machine préférés.

L'emploi du mode RESERVE est tout indiqué :
SHIFT MODE
SHIFT A
A:MID$ («DR G «, 7 AND PEEK &787C,1)
ENTER
MODE
Désormais la pression des touches SHIFT A ENTER provoquera l'affichage de la première lettre du mode angulaire employé. Certains trouveront pauvre cet affichage monolettre. La version étendue suivante leur est destinée :
SHIFT MODE
SHIFT Z
Z:MID$ («DEG.RAD. GRAD», 4*PEEK &787C−867,4)@
ENTER
MODE
La fonction@ a été rajoutée pour éviter d'avoir à presser ENTER après SHIFT Z, elle peut également l'être dans la version précédente. Mais elle présente un effet inattendu si on l'emploie juste après avoir changé de mode angulaire (ce qui est idiot !) ou être revenu en mode RUN.

Exemple :
CLS
RADIAN
SHIFT Z
2+3
L'affichage de 2+3 s'effectue à partir de 22e caractère de la 3e ligne.
ENTER
«2+3» est à présent répété au début de la 1er ligne, le résultat se trouvant à la fin de la 2e. Les prochains caractères frappés au clavier vont s'afficher sur la 3e ligne contenant toujours «2+3».
Je laisse la plume au SHARPENTIER avisé qui va nous dévoiler le pourquoi de cette bizarrerie.

© 1986 H.FAUCON

La particulatité décrite ne semble pas fonctionner sur toutes les versions de PC−1350.

LE CLUB

LISTING 1

```
10:"A":INPUT "X=";X,"Y=";Y:USING
20:R=√(X*X+Y*Y)
30:IF R=0PRINT "R=0_THETA,PHI=_?":END
40:A=(Y=0)+SGN Y
50:C=ACS (X/R)*A
60:PRINT R,C:END
```

LISTING2

```
101:"F"AREAD G:REM _=>S/J
102:H=G:I=INT G:S=I:J=1:F=1:L=0
103:IF ABS (G-S/J)<=1E-9*ABS GRETURN
104:H=1/(H-I):I=INT H:K=S*I+F:F=S:S=K:K=J*I+L:L=J:J=K:
    GOTO 103
110:"RP"INPUT X,Y:USING :R=√(X*X+Y*Y):A=(Y=0)+SGN Y:G=
    ACS (X/R)*A:IF TAN (π/4)<>1PRINT R,G:END
120:G=G/π:GOSUB 102:PRINT R;"--";S;"π/";J:END
```

DOG FIGHT

Encore un listing en code machine qui va vous faire peur, mais je vous l'assure ce jeu vaut bien quelques heures d'efforts. Il frise les 8 Ko et demande donc une carte Ram aux possesseurs de 1350. Mais seuls les cartes 16 Ko permettent de le rentrer avec l'utilitaire déjà publié dans le N° 13. Aux autres d'être astucieux.

Une fois le programme entré en mémoire, et vérifié grâce aux checksums, sauvez-le sur k7 par :
CSAVE M «D−FIGHT» ; &4080, &5DBF
On le rechargera par la suite, par un CLOAD M «D−FIGHT»

COMMENT JOUER ?

Taper CALL &4080
Appuyer sur la touche [1]
Si vous êtes en Mode RUN le jeu est sonorisé (attention aux profs !!), tandis qu'en Mode PRO vous pouvez vous adonner à cette coupable pratique sans éveiller de soupçons.
Vous êtes au commande d'un vaisseau que vous manipulez dans les 4 directions Haut, Bas, Gauche et Droite par les touches :
[9][.][4] et [*]
Le tir s'effectue avec la touche [SML]
Le but est évidemment la destruction des vaisseaux qui vous agressent. Un bouclier (SHIELD) vous protège temporairement, car il est progressivement détruit par les rafales ennemies. Vous controlez à droite de l'écran son niveau, ainsi que votre SCORE, le nombre de vaisseau restant (LEFT), et le niveau atteint (STAGE).
Au bout d'une minute, le vaisseau mère apparait. A sa première destruction il se déploie en trois vaisseaux. Il faut les détruire pour passer au STAGE suivant. Tous les 100 points votre courage est récompensé par un vaisseau supplémentaire. Le message MISS SHOT indique la destruction du vaisseau mère et BONUS vous

gratifie de points supplémentaires au score. Vous saurez tout quand je vous aurais dit que la touche [S] gèle le jeu et que [C] le reprend. (Ouf !! ça fait du bien). Tandis que BRK l'arrête complètement. Si un [BRK] fige l'écran, taper [C] pour reprendre le contrôle de votre vaillant vaisseau. Comme d'habitude, il vous reste quelques surprises à découvrir et à affronter.
Amusez-vous bien !!

J.F.V.

```
DUMP:  DOG FIGHT
(c) I/O

4080 0378023BE2822410:50
4088 5FF252125C60005D:CE
4090 025F34035102D5E2:A2
4098 97036D0200E2B524:C4
40A0 262F03FDDF784F66:61
40A8 D5104C660428086B:36
40B0 08390D7945B60352:17
40B8 0235E297036D0200:22
40C0 E2B5025F3424262F:A5
40C8 03FDDF03520295E2:AD
40D0 97035F02DFE27D02:3B
40D8 0F3424262F038A02:4B
40E0 0000031E105FEDD4:51
40E8 00105FECD400105F:9E
40F0 A00200003F1F784A:C2
40F8 3D7845C37845F378:E5
4100 46197848BF105FDB:28
4108 D602281011DBD604:D6
4110 2807784639784686:6A
4118 78471B7847C27847:1A
4120 F3784860105FDBD6:33
4128 02280C105FE0901A:2F
4130 78495C7849E0784F:85
4138 4D785071784A8078:40
4140 4AE27850F9784E02:B5
4148 105FDBD602285611:B1
4150 DA576703381F11D8:DB
4158 5767043818030383:9B
4160 45DA3411DA570302:9A
4168 8345DA344EC02F03:16
4170 4EC02F1011D95767:F5
4178 0A380B03098345DA:FB
4180 344EFF2F0311DBD6:75
4188 042834105FEC841A:59
4190 048563033808105F:9E
4198 EC841B2C30105FDB:31
41A0 D5042C2911E7D6FF:FB
41A8 3806784FE82C1E11:48
41B0 DC57425267063A15:83
41B8 11DAD603280F11D8:E4
41C0 D607280911D9D60F:DD
41C8 28032C096B087E41:92
41D0 027945B6105FDBD6:96
41D8 0228AD00071052C5:05
41E0 9018105FA0901978:D8
41E8 4C15105FDBD60228:AB
41F0 07784B9778471B78:B3
41F8 47C27848607840A:F6
4200 105FDBD602280C10:66
4208 5FE0901A78495C78:7E
4210 4CD6784F4D785071:6F
4218 784A807851BA7850:8D
4220 F9784E02105FDBD6:E1
4228 082868D6022829D6:97
4230 80285511DA576703:A9
4238 380B03028345DA34:1E
4240 4EFF2F0311D95767:27
4248 0A380B03098345DA:FB
4250 344EFF2F032C2911:19
4258 E7D6FF3806784FE8:A9
4260 2C1E11DC57425267:89
4268 063A1511DAD60328:41
4270 0F11D9D60F280911:20
4278 DBD68038032C096B:0C
4280 087E41E77945B610:32
4288 5FDBD60228E8D680:78
4290 28EB000102038B0C:B0
4298 11D8D50390020000:53
42A0 041E11A2921A9270:83
42A8 0493710111A09019:63
42B0 11A0901893700611:73
42B8 A5901911A5901893:3F
42C0 700611AA90197848:9A
42C8 BF105FDBD6022804:0D
42D0 78471B7847C27848:1B
42D8 60784D2D105FDBD6:72
42E0 02280C105FE0901A:2F
42E8 78495C7849E0784F:85
42F0 4D785071784A8078:40
42F8 4AE27850F9784E02:B5
4300 105FDBD602283F11:9A
4308 DA576703381F11D8:DB
4310 5767043818030383:9B
4318 45DA3411DA570302:9A
4320 8345DA344EC02F03:16
4328 4EC02F1011D95767:F5
4330 0A380B03098345DA:FB
4338 344EFF2F0311D8D6:72
4340 03383A2C2911E7D6:98
4348 FF3806784FE82C1E:36
4350 11DC57425267063A:7F
4358 1511DAD603280F11:21
4360 D8D607280911D9D6:A6
4368 0F28032C096B087E:60
4370 42C67945B6105FDB:C6
4378 D602283C784D8478:FD
4380 47C2784860784F4D:3D
4388 785071784A807850:43
4390 F9784E02105FD9D6:DF
4398 0F280F11DAD60328:32
43A0 0911DCD60138032C:34
43A8 096B087E437C7945:77
43B0 B67843FB7940DE10:13
43B8 5FE85743380A5211:86
43C0 E60206527940E903:E5
43C8 6B02FCE27D020834:06
43D0 035202D4E2825B34:1E
43D8 0300841524784DBC:41
43E0 784E02027F344EFF:CA
43E8 2F032F1B02073402:BB
43F0 FF344EFF2F032F08:E9
43F8 794090035F029FE2:2E
4400 7D00018B130D0F28:60
4408 0C022026024E2602:CC
4410 4F262C1E8A62FF28:D2
4418 060220262C055974:4C
4420 30268B5958640F74:79
4428 30268B59640F7430:51
4430 26035202ECE2828A:57
4438 62FF28068B63053A:BC
4440 0502052C02590300:96
4448 8414248ADB630A38:C6
4450 060220262C090231:B6
4458 260230262C0E62FF:19
4460 2806022026 2C0559:00
4468 743026023026036B:90
4470 027EE27D02023403:1A
4478 5F02A1E2825B3403:F8
4480 00841524784DBC78:B6
4488 4E02027F344EFF2F:81
4490 032F1B0209340352:E1
4498 02DEE2825B340300:D6
44A0 841524784DBC784E:04
44A8 02027F344EFF2F03:36
44B0 2F1B027F344EFF2F:7B
44B8 03036B02FCE27D02:D0
44C0 0534035202EBE282:DF
44C8 5B34030084152478:C7
44D0 4DBC784E02027F34:86
44D8 4EFF2F032F1B0202:CD
44E0 34035F02A4E2825B:FB
44E8 340300841524784D:B9
44F0 BC784E02027F344E:87
44F8 FF2F032F1B8A62FF:66
4500 382F594334020934:76
4508 78500B105FF2D620:2A
4510 2814021734125F60:5A
4518 01DF4E506111DF2F:FE
4520 094EFF2C08021734:D7
4528 4E642F032F252F2A:91
4530 02FF344EFF2F0310:C4
4538 5FE7D6FF3856036C:18
4540 02D4E27D020B3403:79
4548 5202EAE2825B3403:34
4550 00841524784DBC78:B6
4558 4E02027F344EFF2F:81
4560 032F1B105FE75743:3D
4568 34784FE802013478:92
4570 500B105FF2D62028:DA
4578 10021734125F6001:2F
4580 DF4E206111DF2F09:D6
4588 025F344EFF2F032F:43
4590 212F2902013402FF:B1
4598 344EFF2F032F0810:FA
45A0 5FE657424267192A:CA
45A8 025211E957426710:5E
45B0 2803020852372275:55
45B8 0380DB0200821E78:78
45C0 D90C3710781E0200:C4
45C8 001D1F105FE85775:5F
45D0 023A2067053A0302:07
```

```
45D8 04340378023CE27D:50
45E0 020434035202F7E2:6A
45E8 8202002724272F03:28
45F0 2F1137105FE657DA:FD
45F8 105FE7574283C73A:73
4600 1852340378025FE2:5C
4608 7D072F02027F5202:8A
4610 40344EFF2F032D24:44
4618 37035202A5E29710:BC
4620 5FE957844444048E:3D
4628 1A0345834545105F:DE
4630 F85303204411F953:0F
4638 37105FDE57428EC7:72
4640 3804522C41020052:4F
4648 11D8574267053836:5C
4650 524334035F029AE2:A9
4658 7D020586442F0278:F7
4660 4F2F640F90DB5934:E9
4668 0355028EE2820404:54
4670 2F03841A05921A91:12
4678 6000069000041900:13
4680 0102018B0C37105F:41
4688 D8D607388E11DF57:C2
4690 428FC73804522C83:D5
4698 02005211D9574267:3E
46A0 0B3878524334035F:E6
46A8 02B0E27D02038644:E0
46B0 2F02784F2F6407DA:6C
46B8 105FD85783452B02:93
46C0 445934035F029CE2:B3
46C8 82020584442F0204:86
46D0 901A926000105FE0:EB
46D8 821A90C7380B3A06:76
46E0 9261022C04926101:19
46E8 DA91C7380B3A0692:47
46F0 61082C0492610403:93
46F8 0202029014069000:40
4700 0219105FF2D62028:9A
4708 12022034125F6001:3A
4710 DF4E286111DF4E28:1C
4718 2F0B37784F66D501:74
4720 4C66102808105FDB:3C
4728 D4FE2C46105FDBD6:64
4730 01283FD501105FDA:87
4738 574267042A345243:F7
4740 34035F02CFE28204:CF
4748 042F03105FE0901A:2F
4750 030102FF90140490:3D
4758 1B105FF2D6202812:AC
4760 022034125F6001DF:07
4768 4E206111DF4E202F:5C
4770 0B784F66D5084C66:C7
4778 08380B105FE15742:34
4780 67242A0252784F66:36
4788 D5044C6601380B10:DF
4790 5FE1574367083A02:85
4798 52784F66D5024C66:08
47A0 04380B105FE05742:2F
47A8 67572A0252784F66:69
47B0 D5104C6602380B10:EC
47B8 5FE0574367103A02:8C
47C0 5237886000105FDA:BA
47C8 D603382757433488:8E
47D0 5934035F02D0E282:25
47D8 04042F0357424242:57
47E0 42424267652A0552:13
47E8 482C068859784F96:B8
47F0 2F2237886000105F:DF
47F8 D8D6073863574334:1E
4800 885934035F029AE2:F5
4808 82020584442F02F4:76
4810 1400040490180355:1C
4818 028FE28203009059:E1
4820 8414841404841A91:63
4828 70015984142467FF:EC
4830 3827661038049270:13
4838 0166203804927101:C7
4840 6640380493700166:4C
4848 8038E4937101640F:34
4850 94DB069019482C06:98
4858 8859784F6C2F5E37:D8
4860 886000105FD9D60F:15
4868 3855574334885934:70
4870 035F02B0E2820203:7D
4878 84442F02F4140400:05
4880 0290189259660138:34
4888 0890700263652A28:24
4890 6602380890710263:0E
4898 0B3A1D6604380891:9D
48A0 700163292A126608:A7
48A8 380891710163083A:E8
48B0 07069019482C0688:B8
48B8 59784FBE2F503710:A4
48C0 5FDAD603389611D8:C9
48C8 D607389088600011:9E
48D0 DA57433488593403:C0
48D8 5F02D0E28204042F:CC
48E0 03901A896000105F:05
48E8 D857433489593403:BF
48F0 5F029AE282020584:EA
48F8 442F02F414040404:89
4900 821A750190C73A40:E3
4908 7407C72A3BDA7501:F7
4910 91C73A347407C72A:32
4918 2F06D610282A0300:70
4920 0210821B105FF2D6:E6
4928 202812022034125F:21
4930 6001DF4E386111DF:17
4938 4E382F0B00010201:C4
4940 8D0C02FFD82C02C8:68
4948 2F5D67FF3804482C:A2
4950 098859784F967850:0F
4958 0B2F863788600010:EF
4960 5FD9D60F387A5743:69
4968 34885934035F02B0:5D
4970 E282020384442F02:62
4978 04821A907401C72A:96
4980 5C7505C73A57DA91:99
4988 7401C72A507505C7:F7
4990 3A4B105FE7D6FF38:E8
4998 23784FE88859784F:7A
49A0 BE105FF2D6202812:4F
49A8 022034125F6001DF:07
49B0 4E606111DF4E602F:DC
49B8 0B2C23105FDBD502:7B
49C0 D811F2D620281602:11
49C8 0F340200125F6001:17
49D0 DF344E202F036111:25
49D8 DF422F0D482F7537:80
49E0 886000105FD8D607:0C
49E8 3853574334885934:6E
49F0 035F029CE2820205:6B
49F8 84442F0204821A90:29
4A00 7403C72A357505C7:DE
4A08 3A30DA917403C72A:3D
4A10 297505C73A24105F:37
4A18 DBD50211F2D62028:D3
4A20 16020F340200125F:CE
4A28 6001DF344E202F03:14
4A30 6111DF422F0DD82C:D3
4A38 02482F4E3710785E:E4
4A40 0200001D1F037802:BB
4A48 7CE27D105FE98855:10
4A50 488871053A130203:98
4A58 3403550288E28202:7C
4A60 002724272F032D16:E7
4A68 7005493A13020234:43
4A70 0355028CE2820200:4C
4A78 2724272F032D1437:1C
4A80 105FE0901A926000:EB
4A88 11DBD602380A11DC:F3
4A90 5767052A06610478:D0
4A98 4EB4886000105FDA:33
4AA0 D603381957433488:80
4AA8 5934035F02D0E282:25
4AB0 04042F03901A784E:AA
4AB8 87482F1488600010:0A
4AC0 5FD9D60F381C5743:0B
4AC8 34885934035F02B0:5D
4AD0 E282020384442F02:62
4AD8 04901A784E87482F:72
4AE0 1737886000105FD8:7D
4AE8 D607381E57433488:89
4AF0 5934035F029CE282:F1
4AF8 020584442F020490:94
4B00 000218784EB4482F:0B
4B08 1937105FDBD68828:20
4B10 86D6022874D64028:38
4B18 70D620282CD61028:C8
4B20 1611A0574267302A:21
4B28 03523711DBD51000:5D
4B30 0102018B0C3711F8:DB
4B38 835511A257C73804:E5
4B40 43523711DBD52037:E4
4B48 11A2901A035C02CF:8D
4B50 E282030011A15784:F4
4B58 1424906601380370:DA
4B60 0166023803710191:A7
4B68 6604380370016608:84
4B70 38037101105FA290:4E
4B78 1B11A1574267F02A:E7
4B80 03523711DBD54037:C4
4B88 11A2574267663803:54
4B90 523711DBD5803710:11
4B98 5FDBD6882877D670:7D
4BA0 387311D95742670B:A0
4BA8 386B524334035F02:D0
4BB0 B0E27D020386442F:0D
4BB8 0211A2901A926000:51
4BC0 784F2F6403904470:A1
4BC8 04784F2F640F9144:42
4BD0 105FE0821A740283:E4
4BD8 700290C7380B3A06:4C
```

```
4BE0 9261022C04926101:19
4BE8 DA91C7380B3A0692:47
4BF0 61082C0492610406:96
4BF8 90000219105FF2D6:E2
4C00 202812022034125F:21
4C08 6001DF4E286111DF:07
4C10 4E282F0B37105FDA:30
4C18 D603388911DBD670:CC
4C20 3883D688287F8860:A8
4C28 0011DA5743348859:9A
4C30 34035F02D0E28204:D0
4C38 042F03901A11A282:15
4C40 1A750190C73A1274:A7
4C48 04C72A0DDA740291:E3
4C50 C73A06740CC73A1C:A4
4C58 11A2821A740390C7:1D
4C60 3A407407C72A3BDA:FB
4C68 750191C73A347412:C2
4C70 C72A2F11F957DA11:6C
4C78 A4574283C7382752:38
4C80 11F2D62028120220:55
4C88 34125F6001DF4E38:6B
4C90 6111DF4E382F0B88:99
4C98 59784F9678500B2C:B5
4CA0 02C82F7537D82F01:AD
4CA8 11DBD50811F2D620:C2
4CB0 2816020F34020012:97
4CB8 5F6001DF344E202F:70
4CC0 036111DF422F0D88:5A
4CC8 59784F9600010201:BA
4CD0 8D0C784FFC37105F:02
4CD8 A2821A750290C73A:46
4CE0 127402C72A0DDA74:D4
4CE8 0191C73A06740AC7:DE
4CF0 3A1C11A2821A7402:1B
4CF8 90C73A317405C72A:2C
4D00 2CDA750291C73A25:34
4D08 7410C72A2011DBD5:56
4D10 0211F2D620281602:3B
4D18 0F340200125F6001:17
4D20 DF344E202F036111:25
4D28 DF422F0D37886000:7C
4D30 105FD8D607384D57:00
4D38 4334885934035F02:F0
4D40 9AE282020584442F:FC
4D48 02F4140004049018:BA
4D50 906210380D946104:40
4D58 91700163022A0F2C:CC
4D60 0892700263652A12:10
4D68 0690192C1411DD57:34
4D70 425267032804784F:F1
4D78 FC8859784F6C2C02:3E
4D80 482F4837105FDCD6:17
4D88 01283111E0901A90:85
4D90 631838093A057101:6D
4D98 2C03700191631538:E1
4DA0 093A0571012C0370:59
4DA8 01906318280A9163:32
4DB0 15280511DCD50111:16
4DB8 E0901B3767213A3C:C0
4DC0 34037F02E9E2825B:60
4DC8 DA020434340200DA:24
4DD0 84142F0300040490:62
4DD8 18955159262F0402:B2
4DE0 0026105FF2D62028:A5
4DE8 12024034125F6001:5A
4DF0 DF4E106111DF4E10:EC
4DF8 2F0B370300020686:02
4E00 1437001D90106B7B:EE
4E08 18106C5B18003B90:D2
4E10 10700019001D9010:56
4E18 6BEB18106CCB1800:CD
4E20 3B9010704019001D:C1
4E28 90106B9918106C79:B1
4E30 18003B9010720019:7E
4E38 001D90106C091810:5A
4E40 6CE918003B901072:BA
4E48 4019001D90106BB7:38
4E50 18106C9718003B90:0E
4E58 10740019001D9010:5A
4E60 6C2718106D071800:47
4E68 3B9010744019784F:6F
4E70 66D5044C6620380F:58
4E78 784F66D5084C6610:CC
4E80 28054EFF2D083703:E9
4E88 6A028FE282030090:F2
4E90 59844491595A5864:21
4E98 0734027084142F03:77
4EA0 F414915964073402:93
4EA8 00D05A2F02DA2483:DC
4EB0 47DA2637036A028F:7C
4EB8 E27D035202B1E282:CB
4EC0 9259340300023C84:E4
4EC8 142F039059864491:8A
4ED0 595A586407340270:1C
4ED8 86142F0391596407:21
4EE0 6604281B34020584:6C
4EE8 142F030490000418:F6
4EF0 F4198F0204342450:4A
4EF8 4759262F06376403:99
4F00 34020F8414020A84:6D
4F08 142F030490000918:FB
4F10 F4198F0204342450:4A
4F18 4759262F06026B86:EE
4F20 14F4199402043424:13
4F28 504759262F063702:84
4F30 1B34105FF0821AE2:2C
4F38 8258DA84142F0303:81
4F40 0302018414855910:8C
4F48 5FF0841B37020000:27
4F50 FF106B001F020000:9B
4F58 FF106C001F020000:9C
4F60 9F106D001F37107E:00
4F68 00D4003703028345:D8
4F70 3A1EDA34DA34035F:D6
4F78 029AE27D02058644:CC
4F80 2F02F41984440204:0C
4F88 3424262F032F0810:F7
4F90 5FD8574352370301:5E
4F98 83453A1CDA34DA34:3A
4FA0 035F02CFE27D0606:9E
4FA8 2F03F41904040201:4A
4FB0 3424262F032F0810:F7
4FB8 5FDA574352370308:67
4FC0 83453A1EDA34DA34:3C
4FC8 035F02B0E27D0203:78
4FD0 86442F02F4198444:D0
4FD8 02023424262F032F:E3
4FE0 08105FD957435237:73
4FE8 105FE75743345203:79
4FF0 78025EE282052F02:72
4FF8 02005237105FEA90:74
5000 1A00000210910C02:CB
5008 012C0C105FEA901A:3C
5010 00000201910C2A23:ED
5018 900C11E857425278:F8
5020 45C3105FF2D62028:87
5028 12021034125F6001:2A
5030 DF4EFF6111DF4EFF:CA
5038 2F0B0378021EE27D:34
5040 9178504E9078504E:4D
5048 105FEA901B375934:C8
5050 7850555B58640F34:77
5058 038002D4E2820300:C0
5060 020584142F030200:D3
5068 2702043424272F03:DE
5070 37105FE2D6FF280E:93
5078 026E52784F2F6403:1F
5080 390611E352036A02:F4
5088 F6E28211E3574334:1C
5090 0300027084142F03:3F
5098 11E2574384145200:77
50A0 1310552790180490:DB
50A8 19026F841410553B:C2
50B0 9018049019105FE4:A8
50B8 D6FF280C02D85278:AD
50C0 4F2F640311E55210:3D
50C8 554F00119018036A:CA
50D0 02F8E282105FE557:09
50D8 340300027084142F:70
50E0 0311E4574352D264:1A
50E8 7F8414F4148F0211:C1
50F0 34245047DB262F06:25
50F8 371055610027901B:CC
5100 036A02F6E282105F:38
5108 E357433403000270:26
5110 84142F0311E25784:98
5118 14F4140213348F24:18
5120 5046DB262F06025C:2A
5128 8414F414021334A3:8C
5130 245046DB262F0637:27
5138 00105FA2901A9260:AD
5140 0011DBD608380A11:1D
5148 DD57670628066111:41
5150 78515437035202F5:A0
5158 E282036A028FE27D:C1
5160 9259340301020884:B1
5168 142F039159640734:CF
5170 0300022184142F03:F0
5178 91595A5864073402:3D
5180 7086142F03905986:AB
5188 140020049018020A:EC
5190 343434F4198F2450:AC
5198 47DB262F06026586:6A
51A0 14F4199A245047DB:51
51A8 262F0602658614F4:50
51B0 19A5245047DB262F:A9
51B8 0637105FA2901A92:8A
51C0 600011DBD608380A:6C
51C8 11DD576706280661:41
51D0 1178515437000000:65
51D8 00446F6700466967:30
51E0 6874003133353000:A5
51E8 0000000000000043:43
51F0 6F70797269676874:76
51F8 0062790059616D61:63
5200 6461000000000000:C5
```

```
5208 0000000000000000:00
5210 0000000000000000:00
5218 0000000000000000:00
5220 50757368006F6E6C:E9
5228 7900223122006B65:BE
5230 7900000000000000:79
5238 0000000000000000:00
5240 00000053434F5245:7C
5248 0030303030000000:C0
5250 0000000000000000:00
5258 000000534849454C:75
5260 4400000000000000:44
5268 0000000000000000:00
5270 0000004C45465400:2B
5278 0000000000000000:00
5280 0000000000000000:00
5288 0000005354414745:74
5290 0000000000181500:2D
5298 0000000600030000:09
52A0 0000000000150A14:33
52A8 0A13091209110810:6A
52B0 080F070E070D060C:52
52B8 060B050A05090408:3A
52C0 0307020601000065:78
52C8 100000000047414D:E5
52D0 45204F564552204D:0E
52D8 4953532053484F54:4D
52E0 534849454C442042:1B
52E8 4F4E5553200A0503:77
52F0 0201000E041F1F04:57
52F8 1C083E3E0838107C:6C
5300 7C107020F8F820E0:0C
5308 40F0F04000000101:62
5310 00C080E0E0800100:81
5318 0303008000C0C000:06
5320 0301070701000080:93
5328 800007020F0F0206:AF
5330 161F16060C2C3E2C:F3
5338 0C18587C581830B0:48
5340 F8B0306060F06060:48
5348 0001010100C0C0E0:63
5350 C0C0000203020080:07
5358 80C0808001050705:52
5360 010000800000030B:8F
5368 0F0B03041F1F040E:71
5370 083E3E081C107C7C:B0
5378 103820F8F8207040:28
5380 F0F040E000010100:02
5388 0080E0E080C00003:83
5390 03000100C0C00080:04
5398 0107070103008080:13
53A0 0000020F0F02070C:35
53A8 0D1F0D0C181A3E1A:CF
53B0 1830347C34306068:24
53B8 F86860C0D0F0D0C0:D0
53C0 000001000080A0E0:01
53C8 A080010103010100:27
53D0 40C0400003030703:50
53D8 0300808080000606:8F
53E0 0F0606150A110A15:6A
53E8 2A1422142A542844:5E
53F0 2854A8508850A850:44
53F8 A010A05001000100:A2
5400 01A0402040A00201:E4
5408 0201024080408040:C5
5410 0502040205800080:12
5418 00800A0508050AF8:9E
5420 F830B0FEFFFFFFFE:D1
5428 B0801F1F0C0D7FFF:05
5430 FFFF7F0D01000000:8B
5438 0000000000000000:00
5440 F0F06060FCFEFEFE:96
5448 FC60003F3F181BFF:0C
5450 FFFFFFFF1B030000:1A
5458 0000000101010000:03
5460 00E0E0C0C0F8FCFC:30
5468 FCF8C0007F7F3036:18
5470 FFFFFFFFFF360600:37
5478 0000000103030301:0B
5480 0000C0C08080F0F8:68
5488 F8F8F08000FFFF61:BF
5490 6DFFFFFFFFFF6D0C:E1
5498 0000000003070707:18
54A0 03000080800000E0:E3
54A8 F0F0F0E00000FFFF:AE
54B0 C3DBFFFFFFFFFFDB:74
54B8 1801010000070F0F:3F
54C0 0F07000000000000:16
54C8 C0E0E0E0C00000FF:1F
54D0 FF86B6FFFFFFFFFF:36
54D8 B630030301010F1F:1C
54E0 1F1F0F0100000000:4E
54E8 0080C0C0C0800000:40
54F0 FEFE0C6CFFFFFFFF:70
54F8 FF6C60070703031F:FE
5500 3F3F3F1F03000000:DF
5508 0000008080800000:80
5510 00FCFC18D8FFFFFF:E5
5518 FFFFD8C00F0F0606:C0
5520 3F7F7F7F3F0600A0:A1
5528 50A854A854A850A8:E8
5530 54A854A850A854A8:EC
5538 54A8500A152A152A:D4
5540 150A152A152A150A:BC
5548 152A152A150A0514:B6
5550 2A55AA55AA55AA54:7B
5558 2A55AA55AA55AA54:7B
5560 28BF5FAF57AF57AF:01
5568 5FAF57AF57AF5FAF:28
5570 57AF57AF5FFAF5EA:44
5578 F5EAF5FAF5EAF5EA:8C
5580 F5FAF5EAF5EAF5FA:9C
5588 FD111B1FFF040EFF:58
5590 B3550B5698562257:D0
5598 7257C9573E589558:6C
55A0 1B59A9591F5A965A:DF
55A8 FD5A895BF15B665C:49
55B0 CC5C100343434343:47
55B8 4343434343434343:18
55C0 5353505010109090:86
55C8 91918181A1A1A2A2:AA
55D0 2222626263634343:54
55D8 4343434343434343:18
55E0 4343434343434343:18
55E8 4353535050101090:39
55F0 9091918181A1A1A2:98
55F8 A222226262636343:B3
5600 4343434343434343:18
5608 43FF102881818181:7E
5610 8181818181818181:08
5618 8181818181818181:08
5620 8181818181818181:08
5628 8191919090101010:F3
5630 1010101010101010:80
5638 1010101010101010:80
5640 1010101010101010:80
5648 1010101010101010:80
5650 5050535343434343:52
5658 4343434343434343:18
5660 4343636362622222:54
5668 2222222222222222:10
5670 2222222222222222:10
5678 2222222222222222:10
5680 2222A2A2A2A2A2A2:10
5688 A2A2A2A2A2A2A2A2:10
5690 A2A2A2A2A2A2FF10:DB
5698 0343434343434343:D8
56A0 4343434343434343:18
56A8 4343434343434343:18
56B0 4343435353505010:1F
56B8 1010101010101010:80
56C0 1010101010101010:80
56C8 1010101010101010:80
56D0 1010101010101010:80
56D8 1010909091918181:64
56E0 8181818181818181:08
56E8 81818181A1A1A2A2:8A
56F0 2222222222222222:10
56F8 2222222222222222:10
5700 2222222222222222:10
5708 2222222262626262:10
5710 6262626262626262:10
5718 6262626262626262:10
5720 FF40288181818181:EC
5728 8181818181818181:08
5730 8181818181818181:08
5738 81818181A1A1A2A2:8A
5740 2222222222222222:10
5748 2222222222222222:10
5750 2222626263634343:54
5758 5353505010109090:86
5760 9191818181818181:28
5768 8181818181818181:08
5770 FF65046262626262:52
5778 6262626262626262:10
5780 6262626262626262:10
5788 6262222222222222:90
5790 2222A2A2A1A18181:CC
5798 9191909010105050:02
57A0 5353434343434343:38
57A8 6363626222222222:12
57B0 2222222222222222:10
57B8 2222222222226262:90
57C0 63634343434343FF:14
57C8 2828818181818181:56
57D0 8181818181818181:08
57D8 8181818181A1A1A2:69
57E0 A222222222222222:90
57E8 2222222222222222:10
```

```
57F0 222222222222A2A2:10
57F8 A1A1818191919090:86
5800 1010505053534343:EC
5808 4343434343435353:38
5810 5050101010101010:00
5818 1010101010101010:80
5820 1010101010101010:80
5828 9090909090909090:80
5830 9090909090909090:80
5838 90909090FF651922:DF
5840 2222222222222222:10
5848 2222222222226262:90
5850 6363434353535050:92
5858 1010909091918181:64
5860 A1A1A2A222222222:0E
5868 2222222222222222:10
5870 2222222222222222:10
5878 222222222222A2A2:10
5880 A1A1818181818181:48
5888 8181818181818181:08
5890 818181FF15288181:C1
5898 8181818181818181:08
58A0 8181818181818181:08
58A8 8181818181819191:28
58B0 9090101050505353:86
58B8 4343636362622222:54
58C0 2222222222222222:10
58C8 2222626263634343:54
58D0 4343535350501010:EC
58D8 1010101010101010:80
58E0 1010909090909090:80
58E8 9090919181818181:46
58F0 8181919190901010:64
58F8 5050535343436363:92
5900 62622222A2A2A1A1:8E
5908 8181818181818181:08
5910 8181818181818181:08
5918 81FF202881818181:CC
5920 8181818181818181:08
5928 81818181A1A1A2A2:8A
5930 2222626263634343:54
5938 5353505010109090:86
5940 9191818191919090:66
5948 1010505053534343:EC
5950 636362622222A2A2:12
5958 A1A1818181818181:48
5960 8181818181818181:08
5968 8191919090101050:33
5970 5053534343636362:A4
5978 622222A2A2A1A181:AD
5980 8191919090101010:F3
5988 1010101010505053:43
5990 5343435353505010:2F
5998 1090909191818181:D5
59A0 81818181818181FF:86
59A8 5503434343434343:EA
59B0 4343434343434343:18
59B8 4343434343435353:38
59C0 5050101090909191:02
59C8 8181A1A1A2A22222:CC
59D0 2222222222222222:10
59D8 2222222222222222:10
59E0 2222222222222222:10
59E8 2222222222226262:90
59F0 6363434343434343:58
59F8 4343636362622222:54
5A00 A2A2A1A181819191:AA
5A08 9090101050505353:86
5A10 4343434343434343:18
5A18 4343434343FF1503:66
5A20 4343434343434343:18
5A28 4343434343434343:18
5A30 4343434343535350:45
5A38 5010109090919181:33
5A40 81A1A1A2A2222222:6D
5A48 2222222222222222:10
5A50 2222626263634343:54
5A58 4343434343434343:18
5A60 5353505010101010:86
5A68 1010101010101010:80
5A70 1010101010101010:80
5A78 1010909091918181:64
5A80 A1A1A2A222226262:8E
5A88 6363434343434343:58
5A90 43434343FF280343:79
5A98 4343434343434343:18
5AA0 4343434343434343:18
5AA8 636362622222A2A2:12
5AB0 A1A1818191919090:86
5AB8 1010101010101010:80
5AC0 1010505053534343:EC
5AC8 4343434343436363:58
5AD0 62622222A2A2A1A1:8E
5AD8 8181919190901010:64
5AE0 1010101010101010:80
5AE8 1010101090909191:82
5AF0 8181818181818181:08
5AF8 818181FF15034343:20
5B00 4343434343434343:18
5B08 4343434343434343:18
5B10 4343434343434343:18
5B18 4343435353505010:1F
5B20 1010101010101090:00
5B28 9091918181A1A1A2:98
5B30 A222226262636343:B3
5B38 4353535050101090:39
5B40 9091918181818181:37
5B48 81818181818181A1:28
5B50 A1A2A22222222222:8F
5B58 2222222222626263:D1
5B60 6343435353505010:3F
5B68 10909091918181A1:F5
5B70 A1A2A22222222222:8F
5B78 2222222222222222:10
5B80 22222222222222FF:ED
5B88 4028818181818181:6E
5B90 8181818181818181:08
5B98 8181A1A1A2A22222:CC
5BA0 6262636343435353:B6
5BA8 5050101010101010:00
5BB0 1010101090909191:82
5BB8 8181818181818181:08
5BC0 818181818181A1A1:48
5BC8 A2A2222222222222:10
5BD0 2222222262626363:12
5BD8 4343535350501010:EC
5BE0 9090919181818181:46
5BE8 81818181818181FF:86
5BF0 5003626262626262:9F
5BF8 6262626262626262:10
5C00 6262626262626262:10
5C08 62622222A2A2A1A1:8E
5C10 8181919190901010:64
5C18 5050505050505050:80
5C20 5050505050501010:00
5C28 909091918181A1A1:86
5C30 A2A2222262626262:10
5C38 6262626262626262:10
5C40 6262622222222222:D0
5C48 2222A2A2A1A18181:CC
5C50 9191909010105050:02
5C58 5353434343434343:38
5C60 43434343FF652022:B2
5C68 2222222222222222:10
5C70 2222222222222222:10
5C78 2222222222222222:10
5C80 222222A2A2A1A181:6D
5C88 8191919090101050:33
5C90 5053534343636362:A4
5C98 6222222222222222:50
5CA0 2222222222222222:10
5CA8 22222222222222A2:90
5CB0 A2A1A18181919190:98
5CB8 9010105050535343:39
5CC0 4343434343434343:18
5CC8 43434343FF0404FF:12
5CD0 02020A0A0A080809:3B
5CD8 0909010105050504:27
5CE0 0406060602020606:26
5CE8 0604040505050101:1F
5CF0 09090908080A0A0A:49
5CF8 02020A0A0A080809:3B
5D00 0909010105050504:27
5D08 0406060602020606:26
5D10 0604040505050101:1F
5D18 09090908080A0A0A:49
5D20 02020A0A0A080809:3B
5D28 0909010105050504:27
5D30 0406060602020606:26
5D38 0604040505050101:1F
5D40 09090908080A0A0A:49
5D48 02020A0A0A080809:3B
5D50 0909010105050504:27
5D58 0406060602020606:26
5D60 0604040505050101:1F
5D68 09090908080A0A0A:49
5D70 02020A0A0A080809:3B
5D78 0909010105050504:27
5D80 0406060602020606:26
5D88 0604040505050101:1F
5D90 09090908080A0A0A:49
5D98 02020A0A0A080809:3B
5DA0 0909010105050504:27
5DA8 0406060602020606:26
5DB0 0604040505050101:1F
5DB8 09090908080A0A0A:49
```

AMORTISSEMENT

Vous voulez savoir comment amortir votre PC, ou connaitre sa valeur en occasion ? Ce programme est pour vous.

Il sera utile à toute personne désireuse de vendre ou d'acheter un matériel quelconque. Et ceci en tenant compte de la dépréciation de celui-ci, au cours des années d'utilisation. Il peut également servir à une entreprise ou à un artisan, qui veut savoir de combien par an il devra amortir un matériel acquis neuf. Ce qui permettra d'évaluer sa valeur de revente à un instant donné.

Exemple :

Vous avez acheté une voiture 25 000 F (pas de publicité !). Celle-ci perd malheureusement pour vous, de sa valeur au cours des ans. Il vous faut connaitre sa valeur réelle dans 5 ans. Sachant que vous l'avez achetée le 1er Mars 1986, il vous reste aujourd'hui 10 mois pour finir l'année en cours (moment où j'écris ces lignes). A partir de là vous pouvez choisir de l'amortir linéairement (voir ex.) et l'annuité est constante chaque année, ou alors user d'un amortissement dégressif ; la valeur nette comptable de fin devient celle de début.

Mode d'emploi :

Date acquisition : 1/3/86
Valeur d'acquisition : 25 000 F
Durée probable d'utilisation : 5 ans
Nbre de mois pour la 1er annuité : 10 mois
Les résultats figurent sur le listing exemple.
Les options sont claires : L pour linéaire, D pour Dégressif

Abréviations :

V.N.C. Début = Valeur Nette Comptable de début ou valeur d'acquisition
Année Achat = Date de départ de l'Amortissement
Nombre de Mois = Pour la première Annuité
Durée = Durée probable d'utilisation
Tx Line = Taux Linéaire
Tx Deg = Taux Linéaire * Coefficient (selon l'année)

- 1.5 < 5 ans
- 2 5 ou 6 ans
- 2.5 > 6 ans

(Linéaire) Base = valeur d'acquisition
Annuité = annuité ou valeur amortie pour l'année en cours
V.N.C. = valeur de fin d'exercice
(Dégressif) Idem ci-dessus, hormis que la V.N.C. de fin devient celle de début.

Jean-Marc DREVET

```
6000:"C" CLS : WAIT 0:
     CLEAR
6001:PRINT "Amortisseme
     nt": PRINT "------
     -------": PRINT "
     Lineaire  ?":
     PRINT " Degressif
     ?"
6002:CALL &1204:PE=
     PEEK &6F57
6003:IF PE=68 GOSUB "TT
     ":DT=DT-1: GOSUB "
     TX": GOSUB "DIM":
     GOTO "FX"
6004:IF PE<>68 GOSUB "T
     T":DT=DT-1: GOSUB
     "TX": GOSUB "DIM":
     GOTO "LN"
6006:"TX"TL=100/DU
6007:IF DU<5 LET TD=TL*
     1.5
6008:IF DU=5 OR DU=6
     LET TD=TL*2
6009:IF DU>6 LET TD=TL*
     2.5
6010:CLS : CURSOR 0,1:
     PRINT "Taux Lin  "
     ; USING "###.##";T
     L;" %": GOTO 6044
6011:"DIM" DIM AN(DU+1)
     ,FX(DU),VD(DU+1),V
     F(DU+1): RETURN
6012:"LN"AN(1)=BA*TL/10
     0*MO/12:CU=CU+AN(1
     ):VF(1)=BA-CU
6013:FOR I=2 TO DU:AN(I
     )=BA*TL/100:CU=CU+
     AN(I):VF(I)=BA-CU:
     NEXT I
6014:IF VF(1)<>VF(2)
     LET AN(DU+1)=AN(2)
     -AN(1):CU=CU+AN(DU
     +1):VF(DU+1)=BA-CU
6015:FOR K=1 TO DU+1:DT
     =DT+1: CLS
6016:PRINT "Annee  ",
     STR$ DT
6018:PRINT "Debut  ";
     USING "########.##
     ";BA
6020:PRINT "Anuit  ";
     USING "########.##
     ";AN(K)
6021:PRINT "V.n.c. ";
     USING "########.##
     ";VF(K)
6022:CALL &1204
6023:IF VF(K)=0 GOTO "C
     "
6024:NEXT K: GOTO "C"
6025:"TT" CLS : INPUT "
     V.n.c. Debut ";BA,
     "Annee  Achat ";DT
     ,"Nombre Mois  ";M
     O
6026:INPUT "Duree  ";DU
     : RETURN
6027:"DG"AN(1)=BA*TD/10
     0*MO/12:CU=CU+AN(1
     ):VD(1)=BA:VF(1)=V
     D(1)-AN(1)
6028:FOR I=2 TO DU
6029:IF TD>=FX(I) LET V
     D(I)=VF(I-1):AN(I)
     =VD(I)*TD/100:CU=C
     U+AN(I):VF(I)=VD(I
     )-AN(I)
6030:IF TD<FX(I) LET DE
     =I-1: GOTO 6033
6031:NEXT I
6032:"FX"PO=DU+1: FOR Y
     =1 TO DU:PO=PO-1:F
     X(Y)=100/PO: NEXT
     Y: GOTO "DG"
6033:FOR T=I TO DU:AN(T
     )=VF(DE)/(DU-DE)
6034:VD(T)=VF(T-1):CU=C
     U+AN(T):VF(T)=VD(T
     )-AN(T)
6035:NEXT T: FOR N=1
     TO DU:DT=DT+1:
     CLS
6036:PRINT "Annee      "
     ; STR$ DT
6038:PRINT "V.n.c. Deb
     "; USING "########
     .##";VD(N)
6040:PRINT "Annuit
     "; USING "########
     .##";AN(N)
6041:PRINT "V.n.c. Fin
     "; USING "########
     .##";VF(N)
6042:CALL &1204
6043:NEXT N: GOTO "C"
6044:PRINT "Taux Deg  "
     ; USING "###.##";T
     D;" %"
6045:CALL &1204:
     RETURN
```

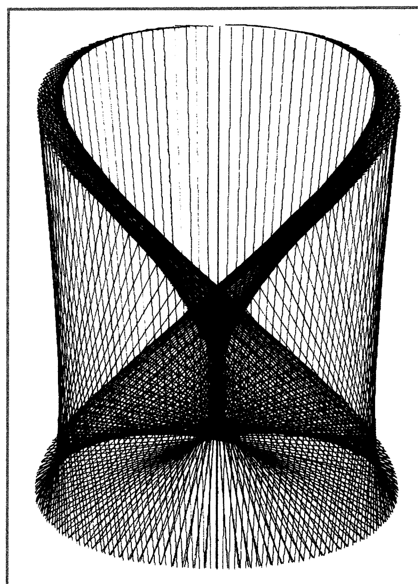
TABLES TRAÇANTES

Vous êtes maintenant nombreux à posséder cette merveille de précision mécanique que sont les tables traçantes MZ 1P01 ou 1P16. Pour exploiter tous leurs talents, il ne leur manquent que vos programmes...

EVASION a un petit goût de vacances que nous vous laissons découvrir. Son avantage est d'exploiter pratiquement toutes les commandes graphiques du basic spécifiques à la table traçante. Son étude détaillée vous permettre de réaliser vos propres créations graphiques... et de nous en faire profiter.

MODULO démontre une fois de plus qu'il est possible de faire beaucoup en très peu de lignes de basic. Ce programme est tiré d'un excellent ouvrage qui vous permettra de créer des dizaines de dessins de ce type sans aucun problème d'adaptation.

DESSINS GEOMETRIQUES ET ARTISTIQUES SUR MICRO-ORDINATEURS
Auteur : Jean-Paul DELAHAYE
Publié aux éditions EYROLLES.



```
10 ' SHARP MZ 700 JARROT D. 7/85
12 '
15 A$="*************** EVASION *******
22 GOSUB3000:' cadre
23 '             1er Bateau
25 XI=240:YI=-360:A=200:B=50
30 TD=180:TF=330:N=30
35 XO=420:YO=-400:L=30
36 U=80:C=0
40 GOSUB2100
42 '             On met les voiles
45 XO=230:YO=-360:L1=300
50 L2=200:U1=90:U2=170:C=1
55 N=30:C=1
60 GOSUB2000
65 XO=380:YO=-375:L1=150
70 L2=320:U1=160:U2=115
75 N=30:C=3
80 GOSUB2000
82 REM Vagues.
85 XD=25:YD=-410:XF=450
90 A=10:D=10:C=1
95 GOSUB2306
100 '          **** MOUETE ****
104 XO=400:YO=-100:AD=90
105 AF=180:R=30:C=0
110 GOSUB2406
115 XO=410:YO=-110:AD=100
120 AF=170:R=40:GOSUB2406
125 XO=350:YO=-100:AD=0
130 AF=90:R=20:GOSUB2406
135 XO=340:YO=-110:AD=5
140 AF=85:R=30:GOSUB2406
145 XO=150:YO=-130:AD=90
150 AF=180:R=25:GOSUB2406
155 XO=160:YO=-140:AD=95
160 AF=175:R=35:GOSUB2406
165 XO=110:YO=-140:AD=5
170 AF=90:R=15:GOSUB2406
175 XO=105:YO=-145:AD=5
180 AF=85:R=20:GOSUB2406
185 XI=100:YI=-260:A=60:B=20
190 TD=180:TF=330:N=20
195 XO=165:YO=-285:L=10
200 U=80:C=0:GOSUB2100
205 XO=100:YO=-265:L1=90
210 L2=60:U1=90:U2=170
215 N=20:C=2:GOSUB2000
220 XO=150:YO=-270:L1=60
225 L2=95:U1=160:U2=115
230 N=20:C=3:GOSUB2000
240 XD=25:YD=-290:A=10:D=10
245 XF=160:C=1:GOSUB2306
250 XD=230:XF=310:GOSUB2306
255 XD=355:XF=455:GOSUB2306
260 XD=370:YD=-350:GOSUB2306
280 XO=90:YO=-50:N=18:A=4
285 DIM TX(N),TY(N)
290 FORI=1TON
295 TX(I)=3*I:TY(I)=A*SIN(I)
300 NEXTI
305 UI=0:UF=340:DU=20:C=3
320 GOSUB2515
325 PHOME:'    **** CERCLE ****
330 FORT=0TO2*PAI(1) STEP .1
345 X=20*COS(T):Y=20*SIN(T)
350 IFT=0THEN MOVEX,Y:GOTO355
352 LINEX,Y
355 NEXTT
360 MOVE-90,50:HSET:MODETN
370 SKIP20:SKIP5:END
2000 '      Enveloppes dans l'angle
2010 '       Conversion degres-radian
2015 U1=PAI(U1)/180:U2=PAI(U2)/180
2025 P1=L1/N:P2=L2/N
2030 R1=0:R2=L2:PCOLORC
2035 FORI=0TON
2040 X1=XO+R1*COS(U1):Y1=YO+R1*SIN(U1)
2045 X2=XO+R2*COS(U2):Y2=YO+R2*SIN(U2)
2050 MOVEX1,Y1
2055 LINE X2,Y2
2060 R1=R1+P1:R2=R2-P2
2065 NEXTI
2070 RETURN
2100 ' Enveloppe=segment=arc ellipse
2120 U=U*PAI(1)/180:TD=TD*PAI(1)
     /180:TF=TF*PAI(1)/180
2130 P=L/N:DT=(TF-TD)/N:R=0:T=TD
2135 PCOLORC:MODEGR
2140 FORI=0TON
2145 X=XO+R*COS(U):Y=YO+R*SIN(U)
2150 XE=XI+A*COS(T):YE=YI+B*SIN(T)
2152 '
2155 MOVEX,Y:LINEXE,YE
2160 R=R+P:T=T+DT
2165 NEXTI
2170 RETURN
2220 T1=T1*PAI(1)/180:T2=T2*PAI(1)/180
2225 V1=V1*PAI(1)/180:VE=VE*PAI(1)/180
2230 PCOLORC
2240 TD=(T2-T1)/N:DV=(V2-V1)/N
2245 T=T1:V=V1
2250 FORI=0TON
2255 X=XO+A1*COS(T):Y=YO+B1*SIN(T)
2260 XE=XI+A2*COS(V):YE=YI+B2*SIN(V)
2265 MOVEX,Y
2270 LINEXE,YE
2275 T=T*DT:V=V+DV
2280 NEXTI
2285 RETURN
2306 PCOLORC
2310 FORX=XDTOXF STEP4
2320 Y=YD+A*SIN(X/D)
2330 IF X=XDTHENMOVEX,Y:GOTO2340
2335 LINEX,Y
2340 NEXTX
2350 RETURN
2406 PCOLORC
2410 AD=AD*PAI(1)/180:AF=AF*PAI(1)/180
2420 FORT=ADTOAF STEP .1
2430 X=XO+R*COS(T):Y=YO+R*SIN(T)
2440 IF T=ADTHENMOVEX,Y:GOTO2450
2445 LINE X,Y
2450 NEXTT
2460 RETURN
2515 UI=UI*PAI(1)/180:UF=UF*PAI(1)
     /180:DU=DU*PAI(1)/180
2520 MOVE XO,YO:HSET:PCOLORC
2530 FOR U=UI TO UF STEP DU
2535 FORI=1TON
2540 RO=SQR(TX(I)*TX(I)+TY(I)*TY(I))
2545 IF(TX(I)=0)*(TY(I)>0)
     THENPHI=PAI(1)/2
2550 IF(TX(I)=0)*(TY(I)<0)
     THENPHI=-PAI(1)/2
2555 IFTX(I)>0 THENPHI=ATN(TY(I)/TX(I))
2560 IFTX(I)<0THENPHI=ATN(TY(I)/TX(I))
     +PAI(1)
2570 X=RO*COS(PHI+U)
2575 Y=RO*SIN(PHI+U)
2580 IF I=1THENMOVEX,Y:GOTO2585
2582 LINEX,Y
2585 NEXTI
2590 NEXTU
2595 RETURN
3000 '          *** Cadre ***
3005 MODEGR:MOVE12,0:PCOLORC
3010 FORI=0TO3
3020 GPRINT[1,I],A$
3025 NEXTI
3030 MOVE405,-440
3040 GPRINT[1,0],"J.D"
3050 RETURN
```

```
20 MODE GR:MOVE0,-900:HSET:NP=480:
   PI=ATN(1)*4
30 N=400:M=N:K1=4:K2=2:H=2
40 DIM X(N),Y(N)
50 FORI=0TON
60 X(I)=INT(NP*.5*(1+SIN(K1*I*PI/N)))
70 Y(I)=INT(NP*.75*(1+COS(K2*I*PI/N)))
80 NEXTI
90 FORI=0TOM
100 I1=I-N*INT(I/N):I2=H*I-N*INT(H*I/N)
110 MOVEX(I1),Y(I1)
120 LINEX(I2),Y(I2)
130 NEXTI
140 END
```

CALC 700

Il manquait un tableur à la panoplie des logiciels utilitaires de nos MZ. Cette lacune est maintenant comblée avec CALC 700, une feuille de calculs électronique qui n'a rien à envier à ses ainées du type « CALC » ou « PLAN ».

CALC 700 est un tableur universel capable de résoudre toutes sortes de problèmes mathématiques, commerciaux, technique ou de gestion. Aucune connaissance en informatique n'est requise pour exploiter pleinement ses possibilités. Le programme fonctionne sur MZ 700 ou 800, sous S. BASIC (BASIC MZ 1Z013).

CALC 700 possède la même structure que ses grands-frères du type CALC, sa rapidité de traitement le situe également au niveau des meilleurs. Son rôle consiste à gérer un tableau de données composé de cases rangées en lignes et colonnes. Ces cases peuvent être reliées les unes aux autres par différents calculs arithmétiques. L'intérêt majeur d'une telle feuille de calculs vient du fait qu'une même opération peut permettre de traiter simultanément et à volonté un nombre important de groupes de cases différents. Il est également possible de répercuter une suite de calculs effectués sur un ensemble de cases sur d'autres ensembles du même type (Modélisation).

Chaque case ou groupe de cases peut être individuellement sauvegardé ou imprimé. La sauvegarde est initialement prévue sur cassette ; une adaptation disquettes reste possible moyennant l'adaptation des routines d'enregistrement-lecture des fichiers.

LES COMMANDES DE CALC 700

T : Affiche les 17 premières lignes à l'écran.
B : Affiche les 17 dernières lignes à l'écran.
SH : Déplacement de colonne affichée, provoque la demande du numéro de colonne.
SR : Déplacement de 1 colonne vers la droite.
SL : Déplacement de 1 colonne vers la gauche.
Z : Déplacement de 1 ligne vers le bas.
L : Effacement total et redéfinition.
LD : Effacement total sans redéfinition.
SO : Tri d'une colonne.
SA : Sauvegarde des données sur cassette.
LO : Lecture des données sur cassette.
HC : Recopie d'écran sur imprimante.
W : Entrée des données d'une colonne.
S : Donne la somme des données d'une colonne.
N : Donne un nom à une colonne.
VZ : Modifie le nombre de colonnes à l'écran (de 1 à 6).
VW : Changement d'une donnée.
RR : Remet le tableau à jour après les commandes VW, W, FE (Répercution sur toutes les données calculées).
FE : Entrée d'une formule de calcul des données d'une colonne, le résultat calculé est inscrit dans la colonne demandée (x).

La formule se compose de 1 ou 2 opérandes.

Pour une opérande, 3 possibilités :
- Un nombre... Il s'inscrit alors sur chaque ligne de la colonne indiquée (x).
- S + No. de colonne (ex.Sn) porte la somme des données de la colonne n dans chaque ligne de la colonne indiquée (x).
- ZS + No. de colonne (ex.ZSn) Porte, à chaque ligne de la colonne indiquée (x), la somme des données des lignes depuis la colonne n jusqu'à la colonne x – 1.

Avec 2 opérandes, le format est : Opérande1 : Opérande2 : Opérateur.
- Le premier opérande ne doit pas être une valeur numérique mais S + No. de colonne ou K + No. de colonne (Sn ou Kn).

Si la lettre est S, la somme de la colonne n est prise comme donnée à chaque ligne.
Si la lettre est K, chaque donnée de la colonne est prise pour la ligne correspondante.
- Le second opérande peut être une valeur numérique ou Sn ou Kn. Le résultat inscrit sur chaque ligne de la colonne indiquée x est le résultat donné en appliquant l'opérateur entre l'opérande1 et l'opérande2. L'opérateur peut être l'un des 4 signes d'opération arithmétique de base + «   » (élévation à la puissance) ou «%» ; ce dernier calculant la différence entre Op1 et Op2/Op1.

STRUCTURE DU PROGRAMME

240-250 Permet de donner un nom à un fichier de données lors de la lecture ou l'enregistrement de ce fichier. Cette routine palie un défaut du BASIC K7 du MZ 700, elle est à supprimer en cas d'utilisation de disquettes.
410-480 Initialisation des variables.
460-600 Initialisation écran et entrée des commandes.
610-840 Partage des taches aux différents sous-programmes.
860-1010 Entrée des données dans la case signalée par un « * ».
1030-1220 Demande du nom d'une colonne. 6 caractères maximum, les autres seront ignorés.
1240-1320 Sous-programme de somme de colonnes affecté aux variables Q (N.C). Si la colonne est affichée, impression à l'écran.
1340-1370 Demande et affiche le nombre de colonnes à l'écran.
1380-1560 Sous programme de modification de valeurs dans une colonne. Si une formule existe déjà, une confirmation est demandée.
1580-1630 Entrée du N° de colonne.
1690-1930 Déplacement de la fenêtre écran. L'exécution reprend à cet endroit après appui sur la touche F3 (Réinitialisation).
1950-2010 La formule entrée est directement prise en compte pour les colonnes respectives (Sous-programme en 2060).
2100-2220 Nouveau calcul sur toutes les colonnes. La ligne 2130 vérifie si seules les données (pas les formules) sont prises en compte.
2240-2880 Calcul des formules. La ligne 2270 vérifie que la formule entrée ne compte qu'un seul chiffre. Les formules à 2 opérandes seront calculées à partir de la ligne 2460.
2600-2680 Saut d'une ligne vers le haut et affichage des valeurs de la dernière ligne.
2690-2790 Affichage de la colonne suivante appelée par « Z ».
2810-2890 Effacement des données, mais pas des formules. Pour éviter toute erreur d'interprétation, les valeurs ne sont pas mises à 0 mais à 0.0001.
3070-3200 Recopie d'écran par lecture du code écran et conversion en code imprimante.

3220-3400 Sauvegarde et lecture des formules et données d'un tableau sur cassette. Les commandes de gestion de fichier séquentiel devront être modifiées pour une utilisation avec disquettes.

Ce programme, tiré et traduit d'une revue allemande est utilisé sans problème depuis plusieurs semaines par notre Sharpentier traducteur et lui donne entière satisfaction. N'hésitez pas à expérimenter toutes les commandes de CALC 700 ; avec un minimum d'expérience, vous pourrez rapidement exploiter les nombreuses possibilités de ce tableur.

```
 REM PROGRAMME 'TABLEUR' pour MZ 731            ******************************
240 POKE18242,205:POKE18252,175
250 POKE18167,205,153,48:POKE12441,17,11,67,33,252,15,205,200,22,58,252,15,201
260 CLS:CLR:ON ERROR GOTO 3040
280 CURSOR 5,5:PRINT[0,6]" C A L C  7 0 0 "
290 CURSOR 5,6:PRINT[0,6]" H. R";CHR$($A8);CHR$($AE);"ner 1984 "
300 CURSOR 5,7:PRINT[0,6]"Tel. 02402/6118 "
310 CURSOR1,14:PRINT"DEFINISSEZ VOTRE FEUILLE DE TRAVAIL"
320 PRINT:PRINT"APPUYEZ UNE TOUCHE S.V.P."
330 GET I$:IF I$="" THEN 330
340 CLS
350 PRINT[0,6]"DEFINITION DE LA FEUILLE DE TRAVAIL";SPC(5)
360 CURSOR 21,2:PRINT".":CURSOR0,2:INPUT"Nombre de lignes ?   ";N1$:N1=VAL(N1$):
IF(N1<1)+(N1>99)GOTO340
370 CURSOR 21,4:PRINT".":CURSOR 0,4:INPUT"Nombre de colonnes ? ";M$:M=VAL(M$):IF
(M<1)+(M>99)THEN CURSOR 21,4:PRINTSPC(9):GOTO370
380 CURSOR 21,6:PRINT".":CURSOR0,6:INPUT"Colonnes a l'ecran ? ";C1$:C1=VAL(C1$):
IF(C1<1)+(C1>6)+(C1>M)THEN CURSOR21,6:PRINT SPC(9):GOTO380
390 CURSOR 21,8:PRINT".":CURSOR 0,8:INPUT"Nom de la feuille de travail ? ";AN$:I
F AN$="." THEN AN$=""
400 CLS:ON-(N1*M*6>SIZE)GOTO 350:N=N1+1
410 DIM C(M),F$(M),N$(M),Q(N,M),V(255)
420 FOR I=0 TO 255:READ V(I):NEXT:RESTORE
430 FOR I=1 TO M:F$(I)="          ":NEXT
440 DEF FNQ(R,C)=INT(Q(R,C)*100+.5)/100
450 I$="":T=0:BR=40
460 FOR V=1 TO C1
470 C(V)=(V*INT(BR/C1)-INT(37/C1)+2)
480 NEXT
490 IF I$="VS" GOTO 1740
500 J=0:K=0:S=0
510 ON-(I$="T")GOTO1740:L=N1
520 S=0:IF L>17 THEN L=17
530 CURSOR 0,3:FOR I=1 TO L:S=S+1:PRINTCHR$(20);S:NEXT
550 REM**ENTREE DES COMMANDES**                         ************************
570 CURSOR 0,0:PRINT[0,6]" COMMANDE ?    ";N1;" Lignes";"   ";M;" Colonnes  ";
580 CURSOR 0,1:FOR I=0 TO 39:PRINT"=";:NEXT
590 FOR C=1 TO C1:CURSOR(C*INT(BR/C1))-INT(21/C1),1:PRINT" C";C-T;" ":NEXT
600 IF I$="LD" GOSUB 1150
610 GOSUB 1670:INPUT"Votre commande ? ";I$
620 ON-(I$=".")GOTO 610:GOSUB 1630
630 IF(I$="T")*(N1<18)+(I$="B")*(N1<18) GOTO 570
640 IFI$="B"GOTO1730:'Aff.17dern.lignes
650 IFI$="T"GOTO500:'Aff.17 prem.lignes
660 IFI$="SL"GOTO1720:'←— de 1 colonne
670 IFI$="SR"GOTO1710:'—→ de 1 colonne
680 IFI$="SH"GOTO1900:'Depla/choix/col.
690 IFI$="Z"GOTO2690:' ↓ de 1 ligne
700 IFI$="SO"GOTO2930:'Tri
710 IFI$="FE"GOTO1970:'Entree/formule
720 IFI$="RR"GOTO2120:'Repercut./resul.
730 IFI$="S"GOTO1260:'Somme/colonne
740 IFI$="W"GOTO880:'Entree donnee
750 IFI$="N"GOTO1050:'Nom de colonne
760 IFI$="VS"GOTO1360:'Nb./colonne aff.
770 IFI$="VW"GOTO1410:'Correctio/donnee
780 IFI$="LD"GOTO2830:'Effac/donnees
790 IFI$="L"GOTO240:'Efface/total+redefinition
800 IFI$="HC"GOTO3160:'Copie/imprim.
810 IFI$="SA"GOTO3240:'Sauv.donn./cass.
820 IFI$="LO"GOTO3340:'Lect.donn./cass.
830 IF I$="O" THEN STOP:'Fin du progr.
840 GOTO 570
860 ' ENTREE DES DONNEES
870 '----------------------------------------------
880 GOSUB 1600
890 CURSOR 0,0:PRINT[0,6]" ENTREE DES DONNEES
900 V=C+T:F$(C)="ZAHL"
910 FOR R=1 TO N1
920 R1=R-K:IF R1<1 GOTO 940
930 CURSOR C(V),R1+2:PRINT"*"
940 GOSUB 1670:INPUT"Donnee  ? ";W$
950 IF W$="." GOTO 940
960 Q(R,C)=VAL(W$):IF R1<1 GOTO 1000
970 GOSUB 1660
980 IF (R1>16)*(R<N1) THEN CURSOR 0,20:PRINT SPC(40);:GOSUB 2620
990 CURSOR 0,21:PRINT SPC(40);
1000 NEXT R
```

```
1010 ON -(T1>16) GOTO 1740:GOTO 570
1030 ' NOM DE COLONNE
1040 '-----------------------------------
1050 GOSUB 1600
1060 V=C+T
1070 CURSOR 0,0:PRINT[0,6]"NOM DE LA COLONNE";SPC(23):CURSOR 17,0:PRINT[0,6]C:CU
RSOR 24,0:PRINT[0,6]"max. 6 lettres"
1080 GOSUB 1670:INPUT"Nom       ? ";N$(C)
1090 IF N$(C)="." GOTO 1080
1100 IF LEN(N$(C))>6 THEN N$(C)=LEFT$(N$(C),6)
1110 IF (V<1)+(V>C1) GOTO 570
1120 CURSOR 0,2:PRINT SPC(2):CURSOR C(V),2:PRINTSPC(6):CURSOR C(V),2:PRINT N$(C)
1130 GOTO 570
1150 CURSOR 0,2:PRINT SPC(40)
1160 FOR C=1-T TO C1-T
1170 IF C=M RETURN
1180 IF LEFT$(N$(C),2)="  "GOTO 1210
1190 V=C+T
1200 CURSOR C(V),2:PRINT N$(C)
1210 NEXT C
1220 RETURN
1240 ' SOMME D'UNE COLONNE
1260 GOSUB 1600
1270 V=C+T
1280 Q(N,C)=0
1290 FOR R=1 TO N1:Q(N,C)=Q(N,C)+Q(R,C):NEXT
1300 IF(V<1)+(V>C1) GOTO 1320
1310 CURSOR C(V),20:PRINT SPC(6):CURSOR C(V),20:PRINT FNQ(N,C)
1320 ON -(I$="RR") GOTO 2210:GOTO 570
1330 '-----------------------------------
1340 'CHANGE Nb colonne/ecran
1360 CURSOR 0,0:PRINT[0,6]"NOMBRE DE COL. a l'ecran";SPC(6);"max. 6"SPC(4);
1370 GOSUB 1610:C1=C:GOTO 460
1380 '-----------------------------------
1390 'CORRECTION D'UNE DONNEE
1410 X$="J"
1420 GOSUB 1600
1430 IF (LEFT$(F$(C),2)<>"  ")*(F$(C)<>"ZAHL")THEN CURSOR 0,0:PRINT[0,6]"CHANGEM
ENT DANS COLONNE CALCULEE ? (O/N)";:GOTO 1450
1440 GOTO 1470
1450 GOSUB 1670:INPUT"Reponse ? ";X$
1460 IF X$<>"O" GOTO 570
1470 CURSOR 0,0:PRINT[0,6]"N° de LIGNE":SPC(29)
1480 GOSUB 1670:INPUT"Numero  ? ";R$
1490 R=VAL(R$):IF(R<1)+(R>N1) GOTO 1480
1500 R1=R-K:V=C+T
1510 CURSOR 0,0:PRINT[0,6]"NOUVELLE VALEUR";SPC(25);
1520 GOSUB 1670:INPUT"Donnee  ? ";Q(R,C)
1530 F$(C)="ZAHL"
1540 IF(V<1)+(V>C1)+(R1<1)+(R1>17)GOTO1560
1550 GOSUB 1660
1560 ON-(Q(N,C)<>0)GOTO 1280:GOTO 570
1580 'ENTREE N° COLONNE
1600 GOSUB 1630:CURSOR 0,0:PRINT[0,6]"NUMERO de COLONNE";SPC(23);:CURSOR 25,0:PR
INT[0,6]"de  1 a";M
1610 GOSUB 1670:INPUT"Numero  ? ";C$
1620 C=VAL(C$):IF(C<1)+(C>M) GOTO 1610
1630 CURSOR 0,0:PRINT SPC(40):RETURN
1650 CURSOR 5,12:PRINT"UN PEU DE PATIENCE  S.V.P.":RETURN
1660 CURSOR C(V),R1+2:PRINT SPC(6):CURSOR C(V),R1+2:PRINTFNQ(R,C):RETURN
1670 CURSOR 0,22:PRINTSPC(40):CURSOR 0,22:PRINTTAB(10)".":CURSOR 0,22:RETURN
1690 'DEPLACEMENT DROITE/GAUCHE
1710 ON-(C1-T=>M)GOTO570:T=T-1:GOTO1740
1720 ON-(T=>0) GOTO 570:T=T+1:GOTO1740
1730 K=N1-17:S=N1
1740 CLS
1750 FOR C=1-T TO C1-T
1760 IF C>M GOTO 1860
1770 V=C+T
1780 IF (N$(C)) THEN CURSOR C(V),2:PRINT N$(C)
1790 IF LEFT$(F$(C),2)="  " GOTO 1850
1800 FOR R=ABS(K)+1 TO ABS(K)+L
1810 R1=R-K
1820 CURSOR C(V),R1+2:PRINTFNQ(R,C)
1830 NEXT R
1840 IF(Q(N,C)) THEN CURSOR C(V),20:PRINTFNQ(N,C)
1850 NEXT C
1860 ON-(I$="T")GOTO 520:S=S-L:GOTO 530
1880 'CHOIX DE COLONNE affichee
1900 CURSOR 0,0:PRINT[0,6]"NUMERO DE COLONNE";SPC(23);
1910 GOSUB 1670:INPUT"Numero  ? ";ES$
1920 ES=VAL(ES$):IF (ES<1)+(ES-1>M-C1) GOTO 1910
1930 T=1-ES:GOTO 1740
1950 'FORMULE DE CALCUL
1970 GOSUB 1600
1980 CURSOR 0,0:PRINT[0,6]"FORMULE DE CALCUL";SPC(23);
1990 GOSUB 1670:INPUT"Formule ? ";F$(C)
2000 IF F$(C)="." GOTO 1990
2010 V=C+T
2020 FOR R=1 TO N1
2030 R1=R-K
2040 GOSUB 2260
2050 IF(V<1)+(V>C1)+(R1<1)+(R1>17)GOTO2070
2060 GOSUB 1660
2070 NEXT R
2080 ON-(Q(N,C))GOTO 1280:GOTO 570
2100 'REPERCUTION changement
```

```
2120 FOR C=1 TO M
2130 IF (F$(C)="ZAHL")+(LEFT$(F$(C),2)="  ")GOTO 2210
2140 V=C+T
2150 FOR R=1 TO N1
2160 R1=R-K:X=Q(R,C):GOSUB 2260
2170 IF(V<1)+(V>C1)+(R1<1)+(R1>17)GOTO2190
2180 IFX<>Q(R,C) GOSUB 1660
2190 NEXT R
2200 IF (Q(N,C)) GOTO 1280
2210 NEXT C
2220 GOTO 570
2240 'FORMULE ( CALCUL )
2260 B$=F$(C)
2270 IF (VAL(B$)) THEN Q(R,C)=VAL(B$):RETURN
2280 IF(LEFT$(B$,1)="S")*(LEN(B$)<4)THEN Q(R,C)=Q(N,VAL(RIGHT$(B$,LEN(B$)[illegible]
ETURN
2290 IF RIGHT$(B$,1)="*"THEN G=1:GOTO 2360
2300 IF RIGHT$(B$,1)="+"THEN G=2:GOTO 2360
2310 IF RIGHT$(B$,1)="/"THEN G=3:GOTO 2360
2320 IF RIGHT$(B$,1)="-"THEN G=4:GOTO 2360
2330 IF RIGHT$(B$,1)="↑"THEN G=5:GOTO 2360
2340 IF RIGHT$(B$,1)="%"THEN G=6:GOTO 2360
2350 IF LEFT$(B$,2)="ZS" GOTO 2570
2360 IF LEFT$(B$,1)="S" THEN ZW=N:GOTO 2380
2370 ZW=R
2380 B$=MID$(B$,2,LEN(B$)-3)
2390 F1=VAL(B$):F1$=STR$(F1)
2400 B$=RIGHT$(B$,LEN(B$)-LEN(F1$)-1)
2410 IF LEFT$(B$,1)="K" THEN ZU=R:GOTO 2440
2420 IF LEFT$(B$,1)="S" THEN ZU=N:GOTO 2440
2430 F2=VAL(B$):ON G GOTO 2460,2480,2500,2520,2540
2440 F2=VAL(RIGHT$(B$,LEN(B$)-1)):GOTO 2450
2450 ON G GOTO2470,2490,2510,2530,2550,2560
2460 Q(R,C)=Q(ZW,F1)*F2:RETURN
2470 Q(R,C)=Q(ZW,F1)*Q(ZU,F2):RETURN
2480 Q(R,C)=Q(ZW,F1)+F2:RETURN
2490 Q(R,C)=Q(ZW,F1)+Q(ZU,F2):RETURN
2500 Q(R,C)=Q(ZW,F1)/F2:RETURN
2510 Q(R,C)=Q(ZW,F1)/Q(ZU,F2):RETURN
2520 Q(R,C)=Q(ZW,F1)-F2:RETURN
2530 Q(R,C)=Q(ZW,F1)-Q(ZU,F2):RETURN
2540 Q(R,C)=Q(ZW,F1)↑F2:RETURN
2550 Q(R,C)=Q(ZW,F1)↑Q(ZU,F2):RETURN
2560 Q(R,C)=Q(ZU,F2)/Q(ZW,F1)*100-100:RETURN
2570 Q(R,C)=0
2580 ZW=VAL(RIGHT$(B$,LEN(B$)-2)):FOR I=ZW TO C-1:Q(R,C)=Q(R,C)+Q(R,I):NEXT:RETU
RN
2600 'SCROLL ROUTINE
2620 IF R1+2>N RETURN
2630 K=K+1:S=S+1
2640 CURSOR 0,20:PRINT CHR$(20);S
2650 POKE85,24:PRINT:RETURN
2670 'SCROLLSHIFT
2690 GOSUB 1670:J=J+1
2700 IF(S=N1)+(R=N1) GOTO 570
2710 R=17+J:K=K+1:S=S+1
2720 CURSOR 0,20:PRINT CHR$(20);S;SPC(37):CURSOR 10,22:PRINT" ":GOSUB 2650
2730 FOR C=1-T TO C1-T
2740 IF LEFT$(F$(C),2)="  " GOTO 2780
2750 V=C+T
2760 CURSOR C(V),19:PRINTFNQ(R,C)
2770 IF (Q(N,C)) THEN CURSOR C(V),20:PRINT FNQ(N,C)
2780 NEXT C
2790 GOTO 610
2810 'EFFACEMENT DONNEES
2830 CLS
2840 GOSUB 1650
2850 FOR C=1 TO M
2860 FOR R=1 TO N1:Q(RR,C)=0:NEXT
2870 IF(Q(N,C))THEN Q(N,C)=.0001
2880 NEXT
2890 CLS:GOTO 500
2910 'TRI
2930 GOSUB 1600
2940 IF N1>20 THEN CLS:GOSUB 1650
2960 FORR=1TON1-1:FORJ=R+1TON1
2970 IF Q(R,C)<=Q(J,C) NEXTJ:GOTO3000
2980 TE=Q(R,C):Q(R,C)=Q(J,C):Q(J,C)=TE
2990 IF J<>N1:NEXTJ
3000 NEXT R:GOTO1740
3020 'TRAITEMENT D'ERREURS
3040 IF (ERL=2380)*(ERN=3) THEN RESUME 570
3050 RESUME 1740
3070 'COPIE D'ECRAN
3090 DATA 32,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,8
7,88,89,90,251,205,221,203,209,48,49,50,51,52,53,54,55,56,57,45
3100 DATA 61,59,47,46,44,229,244,236,218,227,226,215,212,230,232,194,193,196,199
,207,202,128,225,254,200,250,95,248,241,247,63,204
3110 DATA 219,220,233,245,58,94,60,91,243,93,64,201,62,252,92,198,223,208,206,21
1,210,255,33,34,35,36,37,38,39,40,41,43,42,222,246,235,234
3120 DATA 195,197,239,240,228,231,238,237,224,253,216,213,242,249,217,214,192,16
1,154,159,156,146,170,151,152,166,175,169,184,179,176,183
3130 DATA 158,160,157,164,150,165,171,163,155,189,162,187,153,130,135,140,188,16
7,172,145,147,148,149,180,181,182,174,173,186,178
3140 DATA 185,168,177,131,136,141,134,132,137,142,191,133,138,143,190,129,139,14
4,127,252,94,198,95,72,67,96,97,98,99,100,101,102,103,104,112
3150 DATA 113,114,115,116,117,118,119,120,121,122,123,124,125,126,105,[illegible]
```

```
8,227,35,35,35,106,107,108,109,62,110,111,253,112,128,221,205,216,206,213,22
1,208,223,193,254,194,233,245,200
3160 BA=$D000:H=0:PRINT/P:PRINT/P AN$
3170 FOR I=0 TO 40*22-1
3180 J=PEEK(BA+I)
3190 PRINT/P CHR$(V(J));:H=H+1:IF H=40 THEN H=0
3200 NEXT:PRINT/P:GOTO 570
3220 'SAUVEGARDE FORMULE/DONNEES
3240 CLS:INPUT"NOM DE FICHIER ? ";DA$
3250 WOPEN DA$
3260 FOR I=1 TO M
3270 FOR J=1 TO N:PRINT/T Q(J,I):NEXT J
3280 PRINT/T F$(I),N$(I)
3290 NEXT I
3300 CLOSE:GOTO 1740
3320 'LECTURE FORMULE/DANNEES
3340 CLS:INPUT"NOM DE FICHIER ? ";DA$
3350 ROPEN DA$
3360 FOR I=1 TO M
3370 FOR J=1 TO N:INPUT/T Q(J,I):NEXT J
3380 INPUT/T F$(I),N$(I)
3390 NEXT I
3400 CLOSE:GOTO 1740
3410 END
```

MZ 700/800

COPY FICHIERS

Voici un utilitaire bien pratique qui vous permettra d'effectuer, sur disquettes, des copies de sauvegarde de vos fichiers séquentiels (BSD) ou à accès direct (BRD). La présentation et les commandes en sont très simples, il suffit de les suivre à la lettre.

L'OCTET LUCKE
LE SHARPENTIER QUI PROGRAMME
PLUS VITE QUE SON OMBRE



```
10 COPY FICHIERS
40 DIMN$(1200)
70 CONSOLE:COLOR,,2,6:CLS:PRINTTAB(7);"C O P Y     F I C H I E R"
100 PRINTTAB(7);"――――――――――――――――――――――――――――"
130 CONSOLE4,20,1,38:CA$="_":CB$="‾":CC$=CHR$($CA):CD$=CHR$($D4):FORI=1TO36:CA$=
CA$+"_":CB$=CB$+"‾":NEXTI
160 GOSUB430:GOSUB480:CU=3:GOSUB1000
165 IFM<>0THEN180
170 GOTO190
180 CURSORCU+5,13:PRINT[0,]NOM$:CURSOR7,16:PRINT[1,]"CE FICHIER N'EST PAS FINI":
GOSUB370:CURSOR7,16:PRINTSPC(25)
190 GOSUB1010:USR(62)
220 CURSOR22,10:PRINT"DISKETTE OBJET"
225 GOSUB370:IFA$=CHR$($43)THEN4000
250 CU=19:GOSUB1000:CURSORCU+5,13:PRINT[0,]NOM$
260 CURSOR3,13:PRINTSPC(16):OB=1
270 GOSUB1010:IFM<>0THEN500
280 USR(62):CURSOR9,13:PRINT[1,]"AUTRE FICHIERS [O/N]        "
310 GETA$:IF(A$<>"O")*(A$<>"N")THEN310
340 IFA$="O"THENCLR:GOTO40
350 CONSOLE:CLS:NEW
370 GETA$:IF(A$<>CHR$(13))*(A$<>CHR$($43))THEN370
400 RETURN
430 COLOR,,2,:CLS:CURSOR1,8:PRINTCB$:CURSOR1,18:PRINTCA$
460 FORI=8TO18:CURSOR1,I:PRINTCC$:CURSOR38,I:PRINTCD$:NEXTI:RETURN
480 CURSOR4,10:PRINT"DISKETTE SOURCE":GOSUB370:RETURN
500 FORI=1TOJ:N$(I)="":NEXTI
520 J=0:GOTO160
1000 CURSORCU,13:PRINT"NOM";SPC(2)"‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾":RETURN
1010 GETA$:IFA$=CHR$(13)THEN1040
1015 IF(M1<>0)+(M<>0)THEN1040
1020 IFA$=""THEN1010
1025 IFOB=1THENNOM$="":OB=0
1030 NOM$=NOM$+A$:CURSORCU+5,13:PRINT[0,]NOM$:GOTO1010
1040 CURSORCU+7,16:PRINT[1,]"PATIENTEZ":GOSUB2000:CURSORCU+7,16:PRINTSPC(9)
1060 RETURN
2000 ONERRORGOTO3000
2010 IFE=1GOTO2220
2020 ROPEN#3,NOM$
2040 J=J+1:INPUT#3N$(J)
2060 IFEOF(#3)GOTO2100
2080 GOTO2040
2100 CLOSE#3:E=1:GOTO2450
2120 IFM1<>0THENK=M1
2130 XOPEN#5,NOM$
2140 J=J+1:K=K+1:INPUT#5(K),N$(J)
2150 IFSIZE<2500THENM=J-1:GOTO2190
2160 IFEOF(#5)GOTO2190
2180 M=0:GOTO2140
2190 CLOSE#5:E=1:GOTO2450
2220 E=0:IFDR=1THEN2330
2240 WOPEN#7,NOM$
2260 FORI=1TOJ
2280 PRINT#7,N$(I)
2300 NEXTI
2320 CLOSE#7:GOTO2450
2330 IFM1<>0THENK=M1
2340 XOPEN#9,NOM$
2360 FORI=1TOJ
2380 K=K+1:PRINT#9(K),N$(I)
2400 NEXTI
2420 CLOSE#9:IFM<>0THENM1=M1+M
2450 J=J-1:K=0:RETURN
3000 IFERN=40PRINT"        CE FICHIER N'EXISTE PAS":GOSUB3140:GOTO160
3020 IFERN=42PRINT"        CE FICHIER EXISTE DEJA":GOSUB3140:GOTO3060
3040 IFERN=63THENDR=1:GOTO2120
3060 PRINT"     JE LE REMPLACE [O/N]"
3080 GETA$:IF(A$<>"O")*(A$<>"N")THEN3080
3100 IFA$="O"DELETENOM$:GOTO2220
3120 GOTO160
3140 USR(62):FORI=1TO1500:NEXTI:RETURN
4000 CLS:FORI=1TOJ:PRINTI,N$(I)
4010 FORX=1TO30:NEXTX
4020 GETA$:IFA$=CHR$($20)THEN4040
4030 GOTO4060
4040 GETA$:IFA$=CHR$($20)THEN4060
4050 GOTO4040
4060 NEXTI
4070 FORI=1TO1500:NEXTI
4100 GOSUB430:GOSUB480:CU=3:GOSUB1000
4110 CURSOR8,13:PRINT[0,]NOM$:GOTO220
```

BASIC 800

Cette rubrique répond « en vrac » aux questions que vous nous posez le plus souvent par courrier ou téléphone. Dans la mesure du possible, nous agrémenterons ces explications de routines ou programmes explicatifs.

• TI $ est une chaîne de caractères interne au BASIC contenant l'heure sous la forme « HHMMSS ». L'heure est mise à « 000000 » à la mise en route du BASIC. Il est possible, grâce aux instructions de traitement de chaînes de caractères, d'extraire séparément les informations de TI$.
(Listing 1)

```
1 REM UNE HORLOGE TRES SIMPLE
10 INPUT"■Quelle heure est-il ? (HHMMSS) : ";T$
20 TI$=T$
30 H$=LEFT$(T$,2)+":"+MID$(T$,3,2)+":"+RIGHT$(T$,2)
40 IF T$=TI$THEN 30:ELSE T$=TI$
50 SYMBOL[1,2]0,30,H$,5,12,0
60 GOTO 30
```

```
1 REM UNE HORLOGE UN PEU PLUS COMPLIQUE
10 INIT"CRT:M1"
20 H=VAL(LEFT$(TI$,2))
30 CURSOR18,1:PRINT"XII":CURSOR26,3:PRINT"I":CURSOR29,6:PRINT"II":CURSOR31,12:PR
INT"III":CURSOR30,17:PRINT"IV":CURSOR26,21:PRINT"V"
40 CURSOR19,23:PRINT"VI":CURSOR11,21:PRINT"VII":CURSOR6,17:PRINT"VIII":CURSOR7,1
2:PRINT"IX":CURSOR10,7:PRINT"X":CURSOR12,3:PRINT"XI"
50 CIRCLE[3]160,100,80
60 CIRCLE[3]160,100,76
70 A$=TI$:H1=H:M1=M:S1=S
80 DEF KEY( 8)="CONT"
90 H=VAL(LEFT$(TI$,2))
100 M=VAL(MID$(TI$,3,2))
110 S=VAL(RIGHT$(TI$,2)):IF(S=0)*(M<>0)GOSUB320
120 LINE[0]160+SV/40,100-SH/40,160+SH,100+SV
130 LINE[0]160-SV/40,100+SH/40,160+SH,100+SV
140 IFM<>M1THENLINE[0]160+MV/30,100-MH/30,160+MH,100+MV:LINE[0]160-MV/30,100+MH/
30,160+MH,100+MV
150 IFM<>M1THENLINE[0]160+HV/20,100-HH/20,160+HH,100+HV:LINE[0]160-HV/20,100+HH/
20,160+HH,100+HV
160 HH=(SIN(RAD(H*30+M/2)))*48
170 HV=-(COS(RAD(H*30+M/2)))*48
180 MH=(SIN(RAD(M*6)))*65
190 MV=-(COS(RAD(M*6)))*65
200 SH=(SIN(RAD(S*6)))*65
210 SV=-(COS(RAD(S*6)))*65
220 LINE[3]160+HV/20,100-HH/20,160+HH,100+HV
230 LINE[3]160-HV/20,100+HH/20,160+HH,100+HV
240 LINE[3]160+MV/30,100-MH/30,160+MH,100+MV
250 LINE[3]160-MV/30,100+MH/30,160+MH,100+MV
260 LINE[3]160+SV/40,100-SH/40,160+SH,100+SV
270 LINE[3]160-SV/40,100+SH/40,160+SH,100+SV
280 IF(M=0)*(S=0)THEN FORI=1TOH-12*-(H>12):GOSUB330:NEXT
290 IFA$=TI$GOTO290
300 CURSOR0,0:PRINTLEFT$(TI$,2);":";MID$(TI$,3,2);":";RIGHT$(TI$,2)
310 GOTO70
320 MUSIC"O2V15L5T5SOM6 C":RETURN
330 MUSIC"O2V15L5T5SOM6 A":RETURN
```

• SIZE vous permet de connaître le nombre d'octets mémoire encore disponibles sous BASIC. Pour connaître la taille actuelle de votre programme, il suffit de taper : PRINT 22338-SIZE.

• CMT est le nom de périphérique indiquant le lecteur de cassette. Il n'est malheureusement pas possible de lire sur cassette un programme à partir du DISK BASIC. Le seul remède consistue à transférer ce programme sur disquette à l'aide de l'utilitaire TRANS-UTILITY de la MASTER DISQUETTE.

• RND permet d'obtenir une valeur aléatoire entière pouvant se situer entre 0 et 999999999. Pour obtenir une valeur située entre 0 et X, il suffit d'introduire INT (RND(1)*(X+1)).

• STICK et STRIG gèrent la poignée de jeux. Rappelons que tous les « joysticks » commercialisés sous l'appelation « compatible ATARI » peuvent être connectés sur le MZ 800.

• LET est une instruction inutile qui peut être omise dans tous les cas.

• USING : le basic MZ 800 traite ses valeurs numériques sur 8 chiffres significatifs plus point décimal. Actuellement, aucun Basic Sharp ne va au-delà. Pour des applications de type comptabilité sous basic nécessitant des valeurs plus importantes, il est nécessaire de se procurer le système d'exploitation CP/M ainsi que le basic MICROSOFT (M. BASIC) capable de traiter des valeurs en double précision, soit 16 chiffres significatifs.

• GET a deux vilains défauts : il n'est pas à répétition et ne fait pas apparaître le curseur. La routine suivante élimine ces lacunes et peut être incorporée dans tout programme.
(Listing 2)

```
10 REM FOUR FAIRE UN GET A REPETITION AVEC CURSEUR APPARENT
20 POKE$B1A,$20,6
30 GETRE$
40 IF RE$="*" THEN 70
50 PRINTRE$;
60 GOTO 30
70 REM POUR REVENIR AU GET INITIAL
80 POKE$B1A,60,40
```

• INIT"RAM : si vous possédez une carte RAM FILE et une imprimante, cette fonction devient très puissante et peut vous permettre de gagner beaucoup de temps lors des impressions. Essayez, par exemple INIT"RAM :$8000 ». Cet ordre réserve 32000 octets à l'imprimante. Tous les ordres d'impression enverront les caractères destinés à l'imprimante dans cette réserve appelée également BUFFER ; ceux-ci seront ensuite directement gérés par l'imprimante, indépendamment des taches du MZ. Pour vous en convaincre, chargez maintenant un programme dans votre MZ puis tapez : LIST/P:RUN ; vous constaterez que votre programme fonctionne correctement sur l'écran du MZ et qu'il est, en même temps, listé sur l'imprimante.

• INIT"LPT:MO" chaque ordre d'impression est envoyé directement à l'imprimante sans mémorisation dans la RAM FILE.

• INIT"LPT:SO" permet de faire fonctionner l'imprimante table traçante MZ 1P16. Rappelons que l'imprimante MZ 1P01 du MZ 700 ne fonctionne pas sur MZ 800.

•INIT"LPT:S2" permet le branchement de toute imprimante au standard CENTRONICS. La conversion des codes ASCII standard est effectuée automatiquement par le MZ ; par contre, les caractères graphiques du MZ ne seront pas imprimés. Cet ordre doit être utilisé au début de tout programme ou l'utilisation d'une imprimante « CENTRO » est requise.

Dans certain cas, il vous faudra également régler les 2 interrupteurs centraux de la partie arrière du MZ ; il n'y a pas de réglage déterminé, il faudra essayer chacune des 4 combinaisons et ne plus toucher à ces interrupteurs dès que l'imprimante fonctionnera normalement.

• INIT"LPT:S3,$OA" ce mode est utilisé lorsque l'imprimante « CENTRO » connectée au Mz demande un code différent de $OD pour effectuer un retour-chariot et un saut de ligne. Dans certains cas, il peut s'agir du code $OA ; consultez, pour cela, la notice de votre imprimante.

• PATTERN les principales difficultés rencontrées avec cette instruction semblent provenir d'une mauvaise assimilation des valeurs binaires générées par les codes CHR$. A chacun de ces codes correspond une suite de 8 bits correspondant à 8 points allumés ou éteints selon que ces bits sont à 0 ou à 1.

Ensuite, à l'aide de ce modèle, pour dessiner un « X », il vous suffira d'entrer :

PATTERN 8, CHR$
(129,66,36,24,24,36,66,129)

Ces quelques explications ne représentent qu'un début. Nous tenterons, à l'avenir, de résoudre, dans ces colonnes, tout problème particulier posé par une instruction incomprise ou insuffisamment expliquée dans le manuel. Nous tenterons également d'apporter des solutions aux problèmes de programmation que vous nous soumettrez. En un mot, ... nous attendons votre courrier.

S B.

MODIFICATIONS AUX BASICS

Nous allons ajouter aux divers Basics MZ-800 quelques nouvelles possibilités.
Dans la suite nous désignerons par :
(a) la version QD : 5Z009 VI-0A
(b) la version K7 : IZ016 VI-0A
(c) la version K7 : IZ016 VO-3A

Ces nouvelles possibilités sont :
CTRL A : ramène le curseur en tête de ligne d'écran.
CTRL B : permet de passer d'un clavier muet à un clavier sonore et inversement.
CTRL D : efface la fin d'une ligne de texte, à partir de la position du curseur.
CTRL G : produit un bip.
CTRL J : permet d'accéder au second générateur de caractères (utilisable par SYMBOL).
CTRL L : permet d'écrire les caractères en vidéo inversée.

remarque : CTRL J et CTRL L peuvent être « panachées » c'est-à-dire qu'il est aussi possible d'écrire les caractères du second générateur en vidéo inversée.
Cependant CTRL L ne donnera rien avec SYMBOL.

Pour faire ces modifications, suivre attentivement les trois étapes suivantes :
Première étape : mettre l'ordinateur sous tension et appuyer sur M pour retourner au moniteur. On peut alors charger le Basic :
Version QD : placer le QD contenant le Basic dans le lecteur et faire QC (CR).

```
ØØØØ (a)et(b) | (c)     |                REL    55BØH
55BØ
55BØ 3A1A56   | 3A1A56  |  CTRLB:        LD     A,(SWRK)
55B3 EEØ1     | EEØ1    |                XOR    Ø1H
55B5 321A56   | 321A56  |                LD     (SWRK),A
55B8 C9       | C9      |                RET
55B9
55B9 CDC555   | CDC555  |  CTRLA:        CALL   NOPLT?
55BC CØ       | CØ      |                RET    NZ
55BD 2A821Ø   | 2A821Ø  |                LD     HL,(DPRNT)
55CØ 2EØØ     | 2EØØ    |                LD     L,ØØH
55C2 C34DØ9   | C354Ø9  |                JP     ?CURS
55C5
55C5 3A8F1Ø   | 3A8F1Ø  |  NOPLT?:       LD     A,(INDIC)
55C8 B7       | B7      |                OR     A
55C9 C9       | C9      |                RET
55CA
55CA CDC555   | CDC555  |  CTRLD:        CALL   NOPLT?
55CD CØ       | CØ      |                RET    NZ
55CE 2A821Ø   | 2A821Ø  |                LD     HL,(DPRNT)
55D1 CD22Ø8   | CD29Ø8  |                CALL   DEL2
55D4 D9       | D9      |                EXX
55D5 E5       | E5      |                PUSH   HL
55D6 D9       | D9      |                EXX
55D7 C1       | C1      |  CTRLDØ:       POP    BC
55D8 78       | 78      |                LD     A,B
55D9 B1       | B1      |                OR     C
55DA C8       | C8      |                RET    Z
55DB ØB       | ØB      |                DEC    BC
55DC C5       | C5      |                PUSH   BC
55DD 2A821Ø   | 2A821Ø  |                LD     HL,(DPRNT)
55EØ CDE9Ø6   | CDFØØ6  |                CALL   DELSUP
55E3 18F2     | 18F2    |                JR     CTRLDØ
55E5
55E5 3AF955   | 3AF955  |  CTRLJ:        LD     A,(CTRLJM)
55E8 2F       | 2F      |                CPL
55E9 32F955   | 32F955  |                LD     (CTRLJM),A
55EC C9       | C9      |                RET
55ED
55ED 3AØ156   | 3AØ156  |  CTRLL:        LD     A,(CTRLLM)
55FØ 2F       | 2F      |                CPL
55F1 32Ø156   | 32Ø156  |                LD     (CTRLLM),A
```

Entrer le nom BASIC MZ-5Z009 (CR). Le basic se charge.
Répondre N au message « OK ?(Y/N) ».
Version K7 : placer la cassette contenant le Basic dans le magnétophone.
Taper GE807 (CR) et appuyer sur Play.
Attendre la fin du chargement signalée par une étoile.

Deuxième étape : utiliser la commande M du moniteur pour modifier le Basic. On se souviendra que, par exemple, pour placer les codes 3A puis IA puis 56 dans les octets 67B0, 67B1, 67B2, il faut taper, juste après x qui signale que le moniteur attend une commande, M67B0 (CR). Apparaît alors sur l'écran 67B0 00 suivi du curseur. 00 représente le contenu actuel de l'octet 67B0. On peut alors entrer 3A (CR) ce qui va placer 3A dans 67B0. Apparaît alors 67B1 00 suivi du curseur... Pour arrêter toute modification, faire Shift Break. Le moniteur refuse toute entrée invalide. En cas d'erreur (par exemple, on entre 3B au lieu de 3A), faire Shift Break et recommencer. Il n'est pas nécessaire de réentrer un code valide, il suffit d'appuyer sur CR. On entrera à partir de 67B0 les codes du listing 3A IA 56... en les prenant dans les colonnes (a), (b) ou (c) selon la version du Basic. Le dernier octet doit être dans 6821.

Entrer maintenant les modifications des adresses de saut :

Numéro d'octets :

(a)	(b)	(c)
17DF	17DF	17E6
17EC	17EC	17F3
IC87	IC87	IC8E
5FD4	5FD3	6026
125D	125D	125D
1263	1263	1263
1269	1269	1269
126F	126F	126F
1273	1273	1273

Valeurs à y mettre :

```
CD  F5  55
CD  FF  55
CD  19  56
CD  OE  56
B9  55  BO  55
CA  55
F3  44
E5  55
ED  55
```

```
55F4 C9     | C9     |            RET
55F5
55F5 29     | 29     | CHANG1:    ADD    HL,HL
55F6 CBE4   | CBE4   |            SET    4,H
55F8 3E     | 3E     |            DEFB   3EH
55F9 ØØ     | ØØ     | CTRLJM:    DEFB   ØØH
55FA B7     | B7     |            OR     A
55FB C8     | C8     |            RET    Z
55FC CBDC   | CBDC   |            SET    3,H
55FE C9     | C9     |            RET
55FF
55FF F5     | F5     | CHANG2:    PUSH   AF
56ØØ 3E     | 3E     |            DEFB   3EH
56Ø1 ØØ     | ØØ     | CTRLLM:    DEFB   ØØH
56Ø2 B7     | B7     |            OR     A
56Ø3 28Ø4   | 28Ø4   |            JR     Z,+6
56Ø5 F1     | F1     |            POP    AF
56Ø6 2F     | 2F     |            CPL
56Ø7 18Ø1   | 18Ø1   |            JR     +3
56Ø9 F1     | F1     |            POP    AF
56ØA 23     | 23     |            INC    HL
56ØB D9     | D9     |            EXX
56ØC 77     | 77     |            LD     (HL),A
56ØD C9     | C9     |            RET
56ØE
56ØE 29     | 29     | CHANG3:    ADD    HL,HL
56ØF 3AF955 | 3AF955 |            LD     A,(CTRLJM)
5612 B7     | B7     |            OR     A
5613 3E1Ø   | 3E1Ø   |            LD     A,1ØH
5615 C8     | C8     |            RET    Z
5616 3E18   | 3E18   |            LD     A,18H
5618 C9     | C9     |            RET
5619
5619 3E     | 3E     | CHANG4:    DEFB   3EH
561A ØØ     | ØØ     | SWRK:      DEFB   ØØH
561B B7     | B7     |            OR     A
561C C4F344 | C4F344 |            CALL   NZ,BIF
561F C321ØB | C328ØB |            JP     GETCAR
5622
```

Il est possible aussi de modifier l'un des octets de la zone IOFI-IOFE qui contient le nom du basic, ce qui permettra de l'identifier lors de son chargement. Cette modification n'est pas obligatoire !

Troisième étape : réenregistrement du Basic.
Version QD : placer un nouveau QD dans le lecteur et faire JF162. Répondre alors Y au message « OK ?(Y/N) » ce sui réenregistrera le basic.
Version K7 : Placer une nouvelle cassette dans le magnétophone et faire : GE80A suivi de Record.Play pour réenregistrer.

Bernard Kokanoski



DIRECTORY

L'instruction DIR affiche les types et titres des fichiers contenus dans les trois types de mémoire de masse du MZ 800. Il est cependant parfois nécessaires de connaître la taille en octets de ces fichiers. La routine Basic suivante fournit ce type d'indications. L'appel de la fonction DIR fera d'abord apparaître la directory normale puis une seconde ou seront ajoutées la taille de tous les fichiers concernés.

Jean MILLET

```
10 DIR:NO=$27D0:PRINT"■";
25 N=NO+1:IFPEEK(NO)=0THENEND
30 IFPEEK(NO)=1THENPRINT"OBJ.  ";:ELSEPRINT"BTX.  ";
40 IFPEEK(N)<>13THENPRINTCHR$(PEEK(N));:N=N+1:GOTG40
50 PRINTTAB(25);PEEK(NO+21)*256+PEEK(NO+20);" octets":NO=NO+32:GOTO25
```

LECTURE VARIABLE

Il vous est certainement déjà arrivé, lors de l'écriture d'un long programme en Basic, de vous poser cette angoissante question « Ai-je déjà utilisé cette variable dans mon programme ? »

Comme il n'existe malheureusement aucune commande Basic capable de répondre à cette question, il vous suffira de merger ces quelques lignes au programme en cours d'écriture pour vous rassurer. Cette routine est volontairement écrite avec des numéros de lignes élevés pour ne pas interférer avec le programme en cours. Vous n'aurez plus, ensuite qu'à taper GOTO 65500 pour voir s'afficher le nom de toutes les variables numériques, alphanumériques, tableaux dimensionnés, et fonctions définies par DEF FN.

Profitons-en pour signaler une possibilité du basic non signalée dans le manuel : il est tout à fait possible de définir une fonction à l'aide de DEF FN traitant des chaînes de caractères du type :

DEF FNA(B$) = (LEFT$(B$,3) + MID$(B$,6,5)

Jean MILLET

```
65500 Z9=0:N=PEEK($1074)+256*PEEK($1075)
65501 V=PEEK(N):N=N+1:IF(V=0)+(V=5)*(PEEK(N+1)=$5A)*(PEEK(N+2)=$39)THENEND
65502 IFV=3THENGOSUB65506:PRINTTAB(6);"Long:  ";PEEK(N):N=N+3:GOTO65501
65503 IFV=5THENGOSUB65506:PRINT:N=N+5:GOTO65501
65504 IF(V=133)+(V=131)THENGOSUB65507:GOSUB65506:Z$="":N=N+1:PRINT"(";:FORI=1TOP
EEK(N-1):Z$=STR$(PEEK(N)-1+256*PEEK(N+1))+","+Z$:N=N+2:NEXT:PRINTZ$;"■)":N=L:GOT
O65501
65505 IF(V=69)+(V=67)THENGOSUB65507:PRINT"DEF FN ";:GOSUB65506:PRINT"(";CHR$(PEE
K(N+1));")":N=L:GOTO65501
65506 FORI=1TOPEEK(N):PRINTCHR$(PEEK(N+I));:NEXT:N=N+I:IFFRAC(V/16)*16=3THENPRIN
T"$";:RETURN:ELSERETURN
65507 L=PEEK(N)+256*PEEK(N+1)+N:N=N+2:RETURN
```

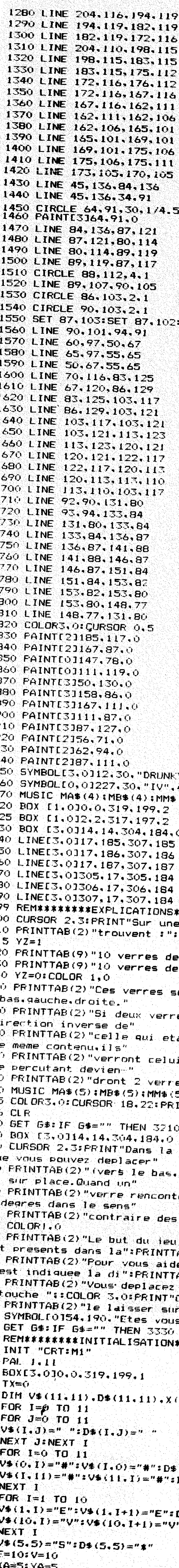
DRUNKY

Ce jeu pourrait se sous-titrer : L'art de mettre du lait dans son vin. Il est écrit entièrement en basic et exploite à fond toutes les commandes graphiques du MZ.

La notice étant intégrée dans le programme (ce qui est une très bonne habitude) nous ne nous étendrons pas sur les règles de ce jeu qui demande une dose assez considérable de patience, de lucidité et de sang froid. Parallèlement à l'intérêt ludique de cet excellent programme, son étude détaillée permettra de clarifier certaines instructions graphiques et sonores dont les possibilités sont insuffisamment détaillées dans le manuel.

Amusez-vous bien, mais n'oubliez surtout pas :« 2 verres, ça va... »

```
[illegible] INIT "CRT:M1":PAL 1.11
100 TEMPO [illegible]
110 DIM MM$(6).MB$(6).MA$(6)
120 MB$(1)="03D7-A7A7E7D7-A7D7-A5-B5D7-A7A7E7D7-A7D7-A5-B5"
130 MM$(1)="02D7-A7A7E7D7-A7D7-A5-B5D7-A7A7E7D7-A7D7-A5-B5"
140 MA$(1)="02D9A9D9D9D9A9D9D9"
150 MB$(2)="03D7-A7A7E7D7-A7D7-A5-B5D7-A7A7E7D7-A7D5D5E5F5G7D7G7-B5-D5D7-A7D7-F5
-G5-A7E7G9D7-A7D5-A5-B5A5"
160 MM$(2)="02D7-A7A7E7D7-A7D7-A5-B5D7-A7A7E7D7-A7D5D5E5F5G7D7G7-B5-D5D7-A7D7-F5
-G5-A7E7G9D7-A7D5-A5-B5A5"
170 MA$(2)="02D9A9D9D9D9A9D9D9G9G9D9D9A9G9D9D9"
180 MB$(3)="03D7-A7A7E7D7-A7D7-A5-B5D7-A7A7G7D7-A7D7-A5-B5D7-A7A7E7D7-A7D7-A5-B5
"
190 MM$(3)="02D7-A7A7E7D7-A7D7-A5-B5D7-A7A7G7D7-A7D7-A5-B5D7-A7A7E7D7-A7D7-A5-B5
"
200 MA$(3)="02D9A9D9D9D9A7G7D9D9D9A9D9D9"
210 MB$(4)="03D7-A7A7G7D7-A7D5D5E5F5G7D7G7-B5C5D7-A7D7-F5-G5-A7C5E5-G7-B5D5D7-A7
D7-A5-B5"
220 MM$(4)="02D7-A7A7G7D7-A7D5D5E5F5G7D7G7-B5C5D7-A7D7-F5-G5-A7C5E5-G7-B5D5D7-A7
D7-A5-B5"
230 MA$(4)="02D9A7G7D9D9G9G9D9D9A9G9D9D9"
240 MB$(5)="03G7D7G7--B5C5D7--A5-#A5--B7-F5-B5-A7E7-G7-B5C5D7-A7A7G7D7-A7D7-A7D7-A7
A7G7A9A8"
250 MM$(5)="02G7D7G7--B5C5D7--A5-#A5--B7-F5-B5-A7E7-G7-B5C5D7-A7A7G7D7-A7D7-A7D7-A7
A7G7A9A8"
260 MA$(5)="02G9G9D9B9A9G9D9A7G7D9D9D9A7G7D7A8"
270 MUSIC MA$(1):MB$(1):MM$(1)
280 BOX[3.0]0.0.319.199.2
290 CIRCLE[3.0]19.179.10.1.3.π*2
300 CIRCLE[3.0]279.179.10.1.3.π*2
310 LINE[3.0]9.129.289.129
320 LINE[3.0]19.189.279.189
330 LINE[3.0]9.129.9.179
340 LINE[3.0]289.129.289.179
350 LINE [3.0]9.157.289.157
360 LINE[3.0]49.129.9.169
370 LINE[3.0]89.129.59.189
380 LINE[3.0]129.129.119.189
390 LINE[3.0]209.129.239.189
400 LINE[3.0]169.129.179.189
410 LINE[3.0]249.129.289.169
420 PAINT[3]39.130.3
430 PAINT[3]119.130.3
440 PAINT[3]199.130.3
450 PAINT[3]279.130.3
460 PAINT[3]59.158.3
470 PAINT[3]159.158.3
480 PAINT[3]259.158.3
490 PAINT[0]1.1.3
500 CIRCLE[1.0]279.19.10.1.0.3.2
510 CIRCLE[1.0]19.19.10.1.0.3.2
520 LINE[1.0]19.9.278.9
530 LINE[1.0]9.129.288.129
540 LINE[1.0]9.19.9.129
550 LINE[1.0]289.19.289.129
560 PAINT[1]100.100.1
570 FOR BO=1 TO 7
580 CIRCLE[1.0]309.70+(BO*15).4.1
590 PAINT[2]309.70+(BO*15).1
600 NEXT BO
610 FOR BO=293 TO 315 STEP 2
620 LINE [2.0]BO.9.BO.69
630 NEXT BO
640 FOR BO=9 TO 69 STEP2
650 LINE[2.0]293.BO.315.BO
660 NEXT BO
670 MUSIC MA$(2):MB$(2):MM$(2)
680 SYMBOL[0.0]32.51."GODEAUX MAURICE".2.2.0
690 SYMBOL[3.0]30.50."GODEAUX MAURICE",2,2,0
700 SYMBOL[0.0]88.91."PRESENTE".2.1.0
710 SYMBOL[3.0]86.90,"PRESENTE",2,1,0
720 MUSIC WAIT
730 MUSIC MA$(3):MB$(3):MM$(3)
740 BOX[1.0]27.39.279.111.1
750 COLOR 0.0
760 LINE 176.136.215.136
770 LINE 176.136.195.127
780 LINE 215.136.201.129
790 LINE 201.129.219.115
800 LINE 219.115.225.110
810 LINE 225.110.230.105
820 LINE 230.105.236.95
830 LINE 236.95.240.85
840 LINE 195.127.199.122
850 LINE 199.122.199.117
860 LINE 199.117.195.112
870 LINE 195.112.197.110
880 LINE 198.109.194.109
890 LINE 194.109.191.99
900 LINE 191.99.186.94
910 LINE 186.94.196.94
920 LINE 196.94.216.92
930 LINE 216.92.226.90
940 LINE 226.90.240.85
950 LINE 186.94.191.91
960 LINE 191.91.206.87
970 LINE 206.87.216.85
980 LINE 216.85.227.83
990 LINE 227.83.232.83
1000 LINE 232.83.240.85
1010 PAINT[2]186.134.0
1020 PAINT[2]230.84.0
1030 CIRCLE 190.106.3.1
1040 CIRCLE 194.100.2.1
1050 CIRCLE 198.100.2.1
1060 PAINT[2]190.106.0
1070 PAINT[3]193.101.0:PAINT[3]197.101.0
1080 LINE194.109.198.107
1090 SET 193.101:SET 194.101:SET 197.101:SET 198.101
1100 LINE 196.88.191.83
1110 LINE 191.83.187.81
1120 LINE 187.81.184.81
1130 LINE 184.81.161.86
1140 LINE 192.91.187.86
1150 LINE 187.86.184.86
1160 LINE 184.85.163.90
1170 LINE 161.86.163.90
1180 LINE 163.90.158.96
1190 LINE 158.96.151.96
1200 LINE 151.96.146.92
1210 LINE 146.92.147.87
1220 LINE 147.87.150.85
1230 LINE 150.85.157.83
1240 LINE 157.83.161.86
1250 SET 150.86:SET 150.87:SET 151.88:SET 152.89:SET 153.89:SET 154.89:SET 154.9
0:SET 154.91:SET 155.91:SET 156.91
1260 LINE 204.110.210.110
1270 LINE 210.110.204.116
```



```
1280 LINE 204.116.194.119
1290 LINE 194.119.182.119
1300 LINE 182.119.172.116
1310 LINE 204.110.198.115
1320 LINE 198.115.183.115
1330 LINE 183.115.175.112
1340 LINE 172.116.176.112
1350 LINE 172.116.167.116
1360 LINE 167.116.162.111
1370 LINE 162.111.162.106
1380 LINE 162.106.165.101
1390 LINE 165.101.169.101
1400 LINE 169.101.175.106
1410 LINE 175.106.175.111
1420 LINE 173.105.170.105
1430 LINE 45.136.84.136
1440 LINE 45.136.34.91
1450 CIRCLE 64.91.30.1/4.5
1460 PAINT[3]64.91.0
1470 LINE 84.136.87.121
1480 LINE 87.121.80.114
1490 LINE 80.114.89.119
1500 LINE 89.119.87.117
1510 CIRCLE 88.112.4.1
1520 LINE 89.107.90.105
1530 CIRCLE 86.103.2.1
1540 CIRCLE 90.103.2.1
1550 SET 87.103:SET 87.102:SET 91.103:SET 91.102
1560 LINE 90.101.94.91
1570 LINE 60.97.50.67
1580 LINE 65.97.55.65
1590 LINE 50.67.55.65
1600 LINE 70.116.83.125
1610 LINE 67.120.86.129
1620 LINE 83.125.103.117
1630 LINE 86.129.103.121
1640 LINE 103.117.103.121
1650 LINE 103.121.113.123
1660 LINE 113.123.120.121
1670 LINE 120.121.122.117
1680 LINE 122.117.120.113
1690 LINE 120.113.113.110
1700 LINE 113.110.103.117
1710 LINE 92.90.131.80
1720 LINE 93.94.133.84
1730 LINE 131.80.133.84
1740 LINE 133.84.136.87
1750 LINE 136.87.141.88
1760 LINE 141.88.146.87
1770 LINE 146.87.151.84
1780 LINE 151.84.153.82
1790 LINE 153.82.153.80
1800 LINE 153.80.148.77
1810 LINE 148.77.131.80
1820 COLOR3.0:CURSOR 0.5
1830 PAINT[2]185.117.0
1840 PAINT[2]167.87.0
1850 PAINT[0]147.78.0
1860 PAINT[0]111.119.0
1870 PAINT[3]50.130.0
1880 PAINT[3]158.86.0
1890 PAINT[3]167.111.0
1900 PAINT[3]111.87.0
1910 PAINT[3]87.127.0
1920 PAINT[2]56.71.0
1930 PAINT[2]62.94.0
1940 PAINT[2]87.111.0
1950 SYMBOL[3.0]12.30."DRUNKY".4.4.0
1960 SYMBOL[0.0]227.30."IV".4.4.0
1970 MUSIC MA$(4):MB$(4):MM$(4)
3020 BOX [1.0]0.0.319.199.2
3025 BOX [1.0]2.2.317.197.2
3030 BOX [3.0]14.14.304.184.0
3040 LINE[3.0]17.185.307.185
3050 LINE[3.0]17.186.307.186
3060 LINE[3.0]17.187.307.187
3070 LINE[3.0]305.17.305.184
3080 LINE[3.0]306.17.306.184
3090 LINE[3.0]307.17.307.184
3099 REM********EXPLICATIONS********
3100 CURSOR 2.3:PRINT"Sur une grille de 10 cases sur 10 se"
3110 PRINTTAB(2)"trouvent :":PRINT
3115 YZ=1
3120 PRINTTAB(9)"10 verres de lait:":II=3:JJ=14:GOSUB"LAIT":PRINT:PRINT
3130 PRINTTAB(9)"10 verres de vin:":II=4.6:JJ=14:GOSUB"VIN":PRINT:PRINT
3140 YZ=0:COLOR 1,0
3150 PRINTTAB(2)"Ces verres se deplacent selon quatre":PRINTTAB(2)"directions:ha
ut.bas.gauche.droite."
3160 PRINTTAB(2)"Si deux verres se heurtent.ils repar":PRINTTAB(2)"tiront dans l
a direction inverse de"
3170 PRINTTAB(2)"celle qui etait la leur.Si.de plus.":PRINTTAB(2)"ces verres son
t de meme contenu.ils"
3180 PRINTTAB(2)"verront celui-ci se modifier.Ainsi.2":PRINTTAB(2)"verres de lai
t se percutant devien-"
3190 PRINTTAB(2)"dront 2 verres de vin."
3200 MUSIC MA$(5):MB$(5):MM$(5)
3205 COLOR3.0:CURSOR 18.22:PRINT"Tapez une touche -->"
3206 CLR
3210 GET G$:IF G$="" THEN 3210
3220 BOX [3.0]14.14.304.184.0
3230 CURSOR 2.3:PRINT"Dans la grille.se trouve egalement.":PRINTTAB(2)"un serveu
r que vous pouvez deplacer"
3240 PRINTTAB(2)"(vers le bas.le haut.la gauche ou la":PRINTTAB(2)"droite)ou lai
sser sur place.Quand un"
3250 PRINTTAB(2)"verre rencontre le serveur sa direc-":PRINTTAB(2)"tion varie de
 90 degres dans le sens"
3260 PRINTTAB(2)"contraire des aiguilles d'une montre":PRINT
3270 COLOR1.0
3280 PRINTTAB(2)"Le but du jeu consiste a n'avoir que":PRINTTAB(2)"des verres de
 lait presents dans la":PRINTTAB(2)"grille.et ce en moins de 100 coups.":PRINT
3290 PRINTTAB(2)"Pour vous aider.vous disposer d'une":PRINTTAB(2)"seconde grille
 ou est indiquee la di":PRINTTAB(2)"rection de chaque verre.":PRINT
3300 PRINTTAB(2)"Vous deplacez le serveur avec le pa-":PRINTTAB(2)"ve curseur ou
 la touche "::COLOR 3.0:PRINT"CR"::COLOR 1.0:PRINT" si vous"
3310 PRINTTAB(2)"le laisser sur place."
3320 SYMBOL[0]54.190."Etes vous pret ?".2.1.0
3330 GET G$:IF G$="" THEN 3330
3339 REM********INITIALISATION********
3340 INIT "CRT:M1"
3350 PAL 1.11
3360 BOX[3.0]0.0.319.199.1
3370 TX=0
3380 DIM V$(11.11).D$(11.11).X(20).Y(20)
3390 FOR I=0 TO 11
3400 FOR J=0 TO 11
3410 V$(I.J)=" ":D$(I.J)=" "
3420 NEXT J:NEXT I
3430 FOR I=0 TO 11
3440 V$(0.I)="#":V$(I.0)="#":D$(0.I)="#":D$(I.0)="#"
3450 V$(I.11)="#":V$(11.I)="#":D$(I.11)="#":D$(11.I)="#"
3460 NEXT I
3470 FOR I=1 TO 10
3480 V$(1.I)="E":V$(1.I+1)="E":D$(1.I)="B":D$(1.I+1)="B"
3490 V$(10.I)="V":V$(10.I+1)="V":D$(10.I)="H":D$(10.I+1)="H"
3500 NEXT I
3510 V$(5.5)="S":D$(5.5)="*"
3520 E=10:V=10
3530 XA=5:YA=5
```

```
3540 FOR I=0 TO 9
3550 X(I)=1:Y(I)=I+1:X(I+10)=10:Y(10+I)=I+1
3560 NEXT I
3570 REM-------------------------------------
3580 REM    AFFICHAGE POSITIONS
3590 REM-------------------------------------
3600 SYMBOL[0] 18,3,"POSITIONS",2,1,0
3610 FOR II=1 TO 10
3620 FOR JJ=1 TO 10
3630 BOX [3,0] 0+((JJ-1)*18),11+((II-1)*15),17+((JJ-1)*18),26+((II-1)*15),0
3640 IF V$(II,JJ)=" " THEN SOUND 60,1
3650 IF V$(II,JJ)="V" THEN GOSUB "VIN"
3660 IF V$(II,JJ)="E" THEN GOSUB "LAIT"
3670 IF V$(II,JJ)="S" THEN GOSUB "SERVEUR"
3680 LINE[3,0]17+((JJ-1)*18),11+((II-1)*15),17+((JJ-1)*18),26+((II-1)*15)
3690 LINE[3,0]18+((JJ-1)*18),11+((II-1)*15),18+((JJ-1)*18),26+((II-1)*15)
3700 NEXT JJ
3710 NEXT II
3720 LINE[1,0] 181,11,181,161
3730 SYMBOL[3]213,5,"DIRECTIONS",1,2,0
3740 FOR II=1 TO 10
3750 FOR JJ=1 TO 10
3760 SOUND 71,1
3770 BOX[0,0]199+((JJ-1)*11),20+((II-1)*11),209+((JJ-1)*11),30+((II-1)*11),3
3780 IF D$(II,JJ)="H" THEN GOSUB"HAUT"
3790 IF D$(II,JJ)="B" THEN GOSUB"BAS"
3800 IF D$(II,JJ)="G" THEN GOSUB"GAUCHE"
3810 IF D$(II,JJ)="D" THEN GOSUB"DROITE"
3820 IF D$(II,JJ)="*" THEN PAINT[2]200+((JJ-1)*11),21+((II-1)*11),0
3830 NEXT JJ:NEXT II
3840 BOX[3,0]0,168,90,199,0
3850 BOX[3,0]91,168,180,199,0
3860 II=12:JJ=1
3870 GOSUB "VIN"
3880 JJ=6:GOSUB"LAIT"
3890 TV$=":"+STR$(V)
3900 SYMBOL[2]12,177,TV$,3,2,0
3910 TE$=":"+STR$(E)
3920 SYMBOL[1]102,177,TE$,3,2,0
3930 GG=1:CC=CC+1
3932 IF V=20 THEN 10000
3934 IF CC>100 AND E<20 THEN 10000
3936 IF E=20 AND CC<=101 THEN 15000
3940 IF TX=0 THEN TI$="000000":TX=1
3950 IF AI$=TI$ THEN 3990
3960 BOX [0,0]185,143,315,163,2:USR(62)
3970 AT$=LEFT$(TI$,2)+":"+MID$(TI$,3,2)+":"+RIGHT$(TI$,2):AI$=TI$
3980 SYMBOL[3]186,150,AT$,2,1,0
3990 IF GG=0 THEN 4040
4000 BOX[3,0]215,168,315,199,0
4010 CO$="COUPS:"+STR$(CC)
4020 SYMBOL[3]220,178,CO$,1,2,0
4030 REM-------------------------------------
4040 GET G$
4050 GG=0
4060 X=XA:Y=YA
4070 IF G$=CHR$(18)THEN X=X-1:GOTO 4130
4080 IF G$=CHR$(17)THEN X=X+1:GOTO 4130
4090 IF G$=CHR$(19)THEN Y=Y+1:GOTO 4130
4100 IF G$=CHR$(20)THEN Y=Y-1:GOTO 4130
4110 IF G$=CHR$(13)THEN 4180
4120 GOTO 3950
4130 IF X<1 THEN X=1
4140 IF X>10 THEN X=10
4150 IF Y<1 THEN Y=1
4160 IF Y>10 THEN Y=10
4170 IF V$(X,Y)<>" " THEN 3950
4180 V$(XA,YA)=" ":V$(X,Y)="S":D$(XA,YA)=" ":D$(X,Y)="*":XA=X:YA=Y
4190 REM-------------------------------------
4200 FOR I=0 TO 19
4210 X=X(I):Y=Y(I)
4220 IF D$(X,Y)="H" THEN X=X-1:GOTO4260
4230 IF D$(X,Y)="B" THEN X=X+1:GOTO 4260
4240 IF D$(X,Y)="D" THEN Y=Y+1:GOTO 4260
4250 IF D$(X,Y)="G" THEN Y=Y-1
4260 IF V$(X,Y)="S" THEN 4330
4270 IF V$(X,Y)<>" " THEN 4390
4280 V$(X,Y)=V$(X(I),Y(I)):D$(X,Y)=D$(X(I),Y(I))
4290 V$(X(I),Y(I))=" ":D$(X(I),Y(I))=" "
4300 X(I)=X:Y(I)=Y
4310 NEXT I
4320 GOTO 3610
4330 USR(62):REM-----------------------------
4340 IF D$(X(I),Y(I))="H" THEN D$(X(I),Y(I))="D":GOTO 4310
4350 IF D$(X(I),Y(I))="B" THEN D$(X(I),Y(I))="G":GOTO4310
4360 IF D$(X(I),Y(I))="G" THEN D$(X(I),Y(I))="H":GOTO 4310
4370 IF D$(X(I),Y(I))="D" THEN D$(X(I),Y(I))="B"
4380 GOTO 4310
4390 REM--------------------------
4400 IF D$(X(I),Y(I))="H" THEN D$(X(I),Y(I))="B":GOTO 4440
4410 IF D$(X(I),Y(I))="B" THEN D$(X(I),Y(I))="H":GOTO 4440
4420 IF D$(X(I),Y(I))="D" THEN D$(X(I),Y(I))="G":GOTO 4440
4430 IF D$(X(I),Y(I))="G" THEN D$(X(I),Y(I))="D"
4440 IF V$(X,Y)<>V$(X(I),Y(I)) THEN 4310
4450 IF V$(X,Y)="V" THEN V$(X,Y)="E":V$(X(I),Y(I))="E":E=E+2:V=V-2::GOTO 4310
4460 V$(X,Y)="V":V$(X(I),Y(I))="V":V=V+2:E=E-2
4470 GOTO 4310
4479 REM********Verre de VIN********
4480 LABEL"VIN"
4490 IF YZ=0 THEN NOISE"O5C1"
4500 POSITION 4+((JJ-1)*18),13+((II-1)*15)
4510 PATTERN [1,0]-13,CHR$($FF,$80,$80,$80,$41,$22,$14,$8,$8,$8,$8,$1C,$3E)
4520 POSITION 12+((JJ-1)*18),13+((II-1)*15)
4530 PATTERN [1,0]-4,CHR$($80,$80,$80,$80)
4540 PAINT[2]5+((JJ-1)*18),14+((II-1)*15),1
4550 RETURN
4559 REM********Verre de LAIT********
4560 LABEL"LAIT"
4570 IF YZ=0 THEN NOISE"O6B1"
4580 POSITION 4+((JJ-1)*18),13+((II-1)*15)
4590 PATTERN [1,0]-13,CHR$($80,$FF,$80,$80,$80,$80,$80,$80,$80,$80,$80,$41,$3E)
4600 POSITION 12+((JJ-1)*18),13+((II-1)*15)
4610 PATTERN [1,0]-11,CHR$($80,$80,$80,$80,$80,$80,$80,$80,$80,$80,$80)
4620 PAINT[3]7+((JJ-1)*18),17+((II-1)*15),1
4630 RETURN
4639 REM********SERVEUR********
4640 LABEL"SERVEUR"
4650 SOUND 36,1
4660 POSITION 4+((JJ-1)*18),12+((II-1)*15)
4670 PATTERN [1,0]-13,CHR$($4,$E,$1F,$3F,$7F,$0,$31,$80,$84,$84,$24,$11,$E)
4680 POSITION 12+((JJ-1)*18),12+((II-1)*15)
4690 PATTERN [1,0]-11,CHR$($0,$0,$0,$80,$C0,$0,$80,$20,$20,$20,$80)
4700 RETURN
4709 REM-------------------------------------
4710 LABEL"HAUT"
4720 SYMBOL[0]200+((JJ-1)*11),22+((II-1)*11),"↑",1,1,0
4730 RETURN
4739 REM-------------------------------------
4740 LABEL"BAS"
4750 SYMBOL[0]200+((JJ-1)*11),22+((II-1)*11),"↓",1,1,0
4760 RETURN
4769 REM-------------------------------------
4770 LABEL"GAUCHE"
4780 SYMBOL[0]200+((JJ-1)*11),21+((II-1)*11),"←",1,1,0
4790 RETURN
4799 REM-------------------------------------
4800 LABEL"DROITE"
4810 SYMBOL[0]200+((JJ-1)*11),21+((II-1)*11),"→",1,1,0
4820 RETURN
10000 CLS:BOX[3,0]69,60,249,108,1
10010 SYMBOL[2,0]79,70,"PERDU",4,4,0
10020 CURSOR 4,20:PRINT"Voulez vous recommencez (O/N) ?"
10030 GET G$
10040 IF G$="N" THEN CLS:END
10050 IF G$="O" THEN CLR:GOTO 3360
10060 GOTO 10030
15000 CLS:BOX[3,0]69,60,249,108,1
15010 SYMBOL[2,0]79,70,"GAGNE",4,4,0:GOTO 10020
```

MZ 80 K

IMPRESSION DE LISTING SOURCE ZEN

Tous les utilisateurs du merveilleux assembleur ZEN X ont certainement remarqué que lorsqu'ils voulaient imprimer leurs travaux sur l'imprimante (option E) que leur cher MZ s'arrêtait et ne redémarrait qu'après un RESET.

Ce phénomène est dû au fait que ZEN a été conçu pour le MZ-700 et celui-ci possède une routine en ROM à l'adresse 018FH nommée trés judicieusement PRINTER. Or dans la ROM de notre cher MZ-80 K il n'y a pas de routine équivalente. Donc, travail fort simple me direz vous, il suffit de « pomper » la routine PRINTER du 700 et de l'adapter au 80K. D'accord mais où la placer ? ZEN se terminant à l'adresse 2599H, la table contenant les étiquettes, commençant à cette même adresse et, se terminant la où commence le programme source, une seule solution : placer la routine en haut de la mémoire.

J'ai donc placé 2 routines à l'adresse 2600H : la première se chargeant de placer la seconde à l'adresse OCFOOH et la seconde étant la fameuse routine d'impression.

Entrez le programme ci-joint puis tapez

```
ZEN>A
OPTION>
ZEN>K
ZEN>A
OPTION>
ZEN>K
ZEN>WO
START>1200H
STOP>2680H
EXEC>2600H
LOAD>1200H
NAME>ZEN X(MZ–80K)
```

Et maintenant à chaque fois que vous chargerez ZEN, une routine d'impression sera placée a l'adresse OCFOOH.

Si lors d'une tentative d'impression, votre imprimante n'est, soit pas connectée, soit pas branchée, ZEN vous dira :

IMPRIMANTE PAS CONNECTEE

P.S. : Pour ceux qui ne disposent pas de 48Ko (et oui, j'ai pensé à eux !) il suffit de modifier la ligne 1 et de rentrer à la place de OCFOOH l'adresse de la fin de votre memoire moins 0100H (256).

Bon LM a tous

A. MORETTE

```
PAGE    1

 1                     IMPOUT:     EQU  0CF00H
 2                                 ORG  2600H
 3                                 LOAD ‡
 4 2600 211426         TRANSPOUT:  LD   HL,FINTRANS+3
 5 2603 1100CF                     LD   DE,IMPOUT
 6 2606 016000                     LD   BC,60H
 7 2609 EDB0                       LDIR
 8 260B 2100CF                     LD   HL,POUT
 9 260E 229E19                     LD   (199EH),HL
10 2611 C30012         FINTRANS:   JP   1200H
11                                 ORG  IMPOUT
12 CF00 F5             POUT:       PUSH AF
13 CF01 3E00                       LD   A,0
14 CF03 CD16CF                     CALL PSTATUS
15 CF06 F1                         POP  AF
16 CF07 D3FF                       OUT  (255),A
17 CF09 3E80                       LD   A,80H
18 CF0B D3FE                       OUT  (254),A
19 CF0D 3E01                       LD   A,1
20 CF0F CD16CF                     CALL PSTATUS
21 CF12 AF                         XOR  A
22 CF13 D3FE                       OUT  (254),A
23 CF15 C9                         RET
24 CF16 C5             PSTATUS:    PUSH BC
25 CF17 D5                         PUSH DE
26 CF18 57                         LD   D,A
27 CF19 1E06                       LD   E,6
28 CF1B 010000                     LD   BC,0
29 CF1E E60D           PST2:       AND  13
30 CF20 BA                         CP   D
31 CF21 C227CF                     JP   NZ,PST3
32 CF24 D1                         POP  DE
33 CF25 C1                         POP  BC
34 CF26 C9                         RET
35 CF27 0B             PST3:       DEC  BC
36 CF28 78                         LD   A,B
37 CF29 B1                         OR   C
38 CF2A C21ECF                     JP   NZ,PST2
39 CF2D 1D                         DEC  E
40 CF2E C21ECF                     JP   NZ,PST2
41 CF31 CD0900                     CALL 9
42 CF34 113DCF                     LD   DE,M1
43 CF37 CD1500                     CALL 15H
44 CF3A C30012                     JP   1200H
45 CF3D 494D5052 M1:               DB   "IMPRIMANTE
                                        PAS CO

45 CF41 4940414E
45 CF45 54452050
45 CF49 41532043
45 CF4D 4F4E4E45
45 CF51 43544545
46 CF55 0D                         DB   13
47                                 END
```